

# Redes Neuronales aplicadas al entrenamiento de Relaciones Constitutivas lineales y no lineales

Mercado N., Freedy.

Facultad de Ingeniería, Universidad de Buenos Aires,  
Av. Paseo Colón 850, CABA, Buenos Aires, Argentina.

23 de septiembre de 2014

## Resumen

En este trabajo se emplean Redes Neuronales Artificiales (RNAs) con entrenamiento supervisado para aprender las relaciones constitutivas de materiales elásticos lineales y no lineales isótropos con miras a emplearlas como sustitutos de las relaciones constitutivas matriciales clásicas en códigos de elementos finitos. Los datos de entrenamiento son generados de forma sintética debido a la ausencia de datos de ensayos experimentales. Las RNAs elegidas son el Perceptrón Simple y el Perceptrón Multicapa, mejorados mediante el algoritmo de Propagación del error hacia atrás (*Backpropagation*). Los resultados muestran un aprendizaje satisfactorio del comportamiento deformación-tensión de los diferentes materiales estudiados.

## 1. Introducción

Las Relaciones Constitutivas (RC) relacionan tensiones, energía libre y flujo de calor con las deformaciones del continuo y la temperatura [3]. En este trabajo limitaremos el alcance a la relación entre tensiones y deformaciones dentro de un material isótropo. Las RC pueden entenderse también desde el punto de vista funcional:

$$\underline{\sigma} = \underline{C}\underline{\varepsilon} \quad (1)$$

Donde  $\underline{\sigma}$  es un vector que almacena las componentes que definen el estado de tensiones,  $\underline{C}$  es la RC matricial clásica de 6 x 6 componentes y  $\underline{\varepsilon}$  el vector que almacena las componentes que definen el estado de deformaciones. Los dos estados descritos y la RC están definidos para un punto de un material.

La Red Neuronal Artificial (RNA) que será empleada es el Perceptrón, el cual está formado por capas, y donde cada capa posee determinado número

de unidades o neuronas. Cada neurona está vinculada a las neuronas de las capas siguientes por medio de una conexión (pesos sinápticos), la cual se caracteriza por modificarse progresivamente conforme se enseñan patrones a la red. A este proceso le llamaremos aprendizaje. El objetivo propuesto en este trabajo es lograr, mediante un método iterativo (*Backpropagation*), la modificación de los pesos sinápticos de toda la red tras la enseñanza repetida de patrones o estados (vectores) de deformaciones y sus correspondientes tensiones. Lo anterior con el fin de reemplazar la RC clásica con una RC definida por la topología de la RN y el valor que tengan sus pesos sinápticos, de tal forma que:

$$\underline{\sigma} = \underline{C}_{RN}\underline{\varepsilon} \quad (2)$$

Donde  $\underline{C}_{RN}$  es la RC dada por la Red Neuronal. Las RN podrían ser útiles para aprender RCs de materiales con comportamientos altamente no lineales, donde las tensiones (salidas de la red) están en función de variables independientes (entradas de la red) que introducen no linealidades en el comportamiento del material, como son la tasa de deformación y el endurecimiento con la deformación para el caso de los metales. Se propone entonces entrenar redes a partir de datos sintéticos que consideraremos producto de ensayos experimentales, y una vez entrenada procederemos a comprobar la calidad de la relación constitutiva encontrada y a tratar de validarla para un conjunto de datos de entrada diferentes a los datos de entrenamiento, es decir, desconocidos para la red.

En la primera parte del artículo se presentará un resumen de la teoría del Perceptrón con *Backpropagation*. Luego, se tratará la temática de las RC y se introducirán el modelo elástico lineal 3D y el modelo elástico lineal 2D para deformación plana, para luego mencionar detalles sobre el pre y post-procesamiento de los datos de entrenamiento, el criterio de conver-

gencia y la topologías de la red. Por último se mostrarán tres aplicaciones del Perceptrón con *Backpropagation* a dos problemas lineales y uno no lineal (RC elasto-plástica).

## 2. Perceptrón Multicapa con Backpropagation

El perceptrón es una Red Neuronal de aprendizaje supervisado que transforma una entrada en una salida. Una topología ilustrativa de un perceptrón multicapa puede ser observada en la Figura 1. El funciona-

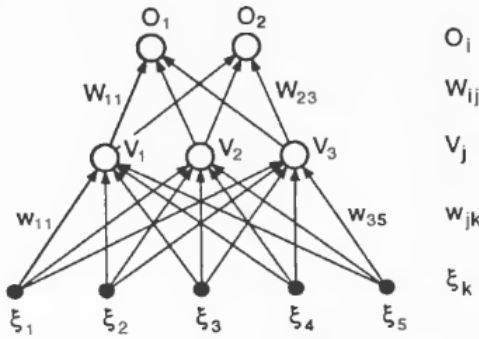


Figura 1: Red Neuronal *Feed-Forward* de 2 capas. Tomada de [8].

miento del Perceptrón, capa por capa, es como sigue: dado un patrón  $\mu$ , la unidad escondida  $j$  recibe una entrada neta  $\xi_k^\mu$

$$h_j^\mu = \sum_k w_{jk} \xi_k^\mu \quad (3)$$

la cual produce la salida

$$V_j^\mu = g(h_j^\mu) \quad (4)$$

La unidad de salida  $i$  recibe la entrada  $V_j^\mu$

$$h_i^\mu = \sum_j W_{ij} V_j^\mu \quad (5)$$

que produce la salida final

$$O_i^\mu = g(h_i^\mu) \quad (6)$$

### 2.1. Función de activación

Para la función de activación  $g(h)$  normalmente se utiliza una función sigmoidea. Esta función debe ser diferenciable y lo normal es desear que se sature en ambos extremos [8]. La función empleada para el caso de estudio no lineal es:

$$g(h) = \tanh \beta h \quad (7)$$

$$\text{para rango } -1 \leq g(h) \leq 1$$

Su derivada es

$$g'(h) = \beta(1 - g^2) = \beta(1 - (\tanh \beta h)^2)$$

El parámetro  $\beta$  a menudo es 1 [8].

### 2.2. Backpropagation

Consiste en corregir los pesos  $w_{ij}$  para cada capa a partir del error entre la salida deseada y la salida actual de la red para un patrón determinado. Los pasos para realizar las correcciones son bastante conocidos y se pueden hallar en la referencia [8].

El algoritmo de *Backpropagation* se aplica para cada patrón presentado a la red hasta presentar todos los patrones. A cada presentación de todos los patrones se le llama un “epoch”. El aprendizaje se extenderá por tantos “epoch” como sea necesario hasta que se cumpla el criterio de convergencia o bien hasta un número de “epochs” determinado por el usuario de la red. Para todos los pesos de la red el algoritmo es:

```

para EPOCH=1 hasta MAXEPOCH
para PATRON=1 hasta NPATRONES
   $\Delta W_{ij} = \Delta W_{ij} + \eta \delta_i V_j$ 
fin
 $W_{ij} = W_{ij} + \Delta W_{ij}$ 
fin

```

## 3. Relaciones constitutivas

### 3.1. Modelo general - 3D

El modelo clásico de la relación constitutiva lineal entre tensiones y deformaciones para el caso más general (3D) está dado por la ecuación 8:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{13} \end{bmatrix} = A \begin{bmatrix} 1 & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{13} \end{bmatrix} \quad (8)$$

$$A = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$$

donde  $\sigma$  y  $\tau$  son tensiones normales y de corte, y  $\varepsilon$  y  $\gamma$  son deformaciones normales y de corte, respectivamente.

### 3.2. Modelo para deformación plana - 2D

Este modelo puede obtenerse a partir de la RC tridimensional de la ecuación 8. En este caso las deformaciones están sobre un plano 2D, más no las tensiones. Esto indica que:

$$\varepsilon_{33} = \varepsilon_{13} = \varepsilon_{23} = 0 \quad (9)$$

$$\sigma_{13} = \sigma_{23} = 0 \quad (10)$$

Simplificando a partir de la Ecuación 8 y teniendo en cuenta que para el caso de deformación plana la componente  $\varepsilon_{33} = 0$  obtenemos (con  $\gamma_{12} = 2\varepsilon_{12}$ ):

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \\ \sigma_{33} \end{bmatrix} = M \begin{bmatrix} 1 - \nu & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \\ \nu & \nu & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} \quad (11)$$

$$M = \frac{E}{(1 + \nu)(1 - 2\nu)} \quad (12)$$

## 4. Entrenamiento del Perceptrón

Para entrenar la red se modifican los pesos de las interconexiones mediante un método iterativo para obtener una señal de salida lo más próxima posible a la salida deseada como respuesta a un patrón de entrada. A este proceso se le llama entrenamiento. Para ello deben conocerse de antemano datos de entrada y salida de la red. Una parte de estos datos puede ser utilizada para entrenarla, mientras que la otra parte puede usarse para realizar pruebas que confirmen un correcto aprendizaje [9].

### 4.1. Procedimiento

El grupo de datos de entrada y salida puede obtenerse a partir de evaluaciones numéricas o ensayos experimentales. Una vez finalizado el entrenamiento, la capacidad de generalización de la red nos permite evaluar tensiones a partir de deformaciones desconocidas para la red. El procedimiento para modelar relaciones constitutivas se divide en tres pasos importantes:

1. Se selecciona la topología de la RNA que pueda representar el problema. El artículo de la Referencia [9] limita esta opción sólo a redes de más de dos capas, sin embargo el artículo de la Referencia [5] logra emplear una red sin capas escondidas para modelar una relación constitutiva elástica lineal.
2. Adquisición de datos que contengan la información que caracteriza el problema muestreando el comportamiento de un material mediante modelación numérica o experimentación real.
3. Inclusión de la RNA como una subrutina en un programa de computadora.

Está comprobado que las RNA son aproximadores universales para cualquier función con muchas variables independientes, donde la información sobre la dependencia funcional es transferida a la RNA por medio de un número de datos discretos que describan el valor de la función para un número suficientemente largo de puntos de muestra en el espacio de las variables de interés. El número de capas escondidas, el número de neuronas y la repartición de pesos entre capas escondidas deben ser definidas por el usuario. Estas características son dependientes del problema y no es posible dar una prescripción general, por lo cual la práctica común es determinar la mejor configuración topológica tratando de minimizar el número de capas ocultas y el número de neuronas por cada capa [1].

### 4.2. Normalización de datos

Antes de entrenar la red tanto las variables de entrada como las de salida deberán ser normalizadas en cierto rango. Dependiendo de la función de activación que utilicemos podemos necesitar normalizar en un rango de 0 a 1, tal como recomiendan las referencias [14] y [10]. Esto nos llevaría a usar un método de normalización ampliamente utilizado, que es

$$X' = \frac{X - 0,95X_{min}}{1,05X_{max} - 0,95X_{min}} \quad (13)$$

donde  $X$  son los datos originales, y  $X_{min}$  y  $X_{max}$  son los valores mínimos y máximos de  $X$ , respectivamente.  $X'$  son los datos unificados de los correspondientes  $X$ . Este procedimiento de pre-procesamiento puede hacer más eficiente el entrenamiento de la red neuronal [13].

Otra forma de normalizar los datos de entrada es la propuesta por Hashash, Jung y Ghaboussi en la Referencia [7]. Se emplea cuando los datos corresponden a

deformaciones que serán empleadas en la capa de entrada de la red neuronal. El vector de deformaciones sería escalado entonces por un factor apropiado:

$$\varepsilon_i^{NN} = \frac{\varepsilon_i}{S_i^\varepsilon} \quad \text{tal que} \quad -1 < \varepsilon_i^{NN} < 1 \quad (14)$$

Los datos de salida también deben ser escalados de vuelta. Siguiendo la notación de la Referencia [7] tendríamos:

$$\sigma_i = S_i^\sigma \sigma_i^{NN} \quad \text{donde} \quad -1 < \sigma_i^{NN} < 1 \quad (15)$$

### 4.3. Criterio de Convergencia

El criterio de convergencia para la red está determinado por el promedio de la raíz media cuadrática del error (RMS por sus siglas en inglés) entre los valores deseados y los de salida, y está dado como,

$$E_{RMS} = \frac{1}{N} \sum_{i=1}^N \sqrt{\frac{1}{p} \sum_{j=1}^p (d_{ij} - y_{ji})^2} \quad (16)$$

donde  $E_{RMS}$  es el *RMS* promedio,  $N$  es el número de datos de entrenamiento o prueba,  $p$  es el número de variables en la salida,  $d_j(n)$  y  $y_j(n)$  son la salida objetivo y la salida de la red para la neurona  $j$ , respectivamente [10]. Este criterio fue el elegido para este trabajo, de modo que el proceso iterativo de aprendizaje será detenido cuando el error  $E_{RMS}$  sea inferior al error permitido.

### 4.4. Cantidad óptima de neuronas en capas ocultas

El número óptimo de neuronas debe reducirse al mínimo para prevenir la tendencia que tiene la red a sobre-ajustarse a los datos [4]. Es posible implementar un algoritmo de adaptación del número de neuronas, el cual inicia con pocas neuronas y las va incrementando a medida que se requiera un mejor ajuste de los pesos a los patrones de entrada y salida. Ver referencia [6] para más información al respecto.

### 4.5. Velocidad de convergencia

Para mejorar este aspecto del entrenamiento del perceptrón se utilizará la introducción de un término de momentum para el cálculo del  $\Delta w_{pq}(t+1)$ . La idea es darle a cada conexión  $w_{pq}$  una inercia o momentum de forma que tienda a cambiar en la dirección de descenso que sienta más “fuerza”, en lugar de oscilar con pequeñas variaciones. Este esquema se implementa

dándole a cada  $\Delta w_{pq}(t+1)$  una contribución de peso proveniente del paso de tiempo anterior:

$$\Delta w_{pq}(t+1) = -\eta \frac{\partial E}{\partial w_{pq}} + \alpha \Delta w_{pq}(t) \quad (17)$$

El parámetro de momentum  $\alpha$  debe estar entre 0 y 1. A menudo se utiliza un valor de 0.9 [8].

## 5. Ejemplos numéricos

### 5.1. Material elástico lineal: RNA entrenada con todas las componentes del modelo de material 2D

El objetivo principal de éste y de los siguientes dos capítulos es comprobar la posibilidad de desarrollar un modelo constitutivo a partir de datos obtenidos mediante mediciones experimentales. En lugar de emplear resultados de pruebas reales se generarán datos sintéticos, es decir, números generados por computadora para el entrenamiento. Esto debido a la falta de datos experimentales reales.

#### 5.1.1. Entrenamiento con todas las componentes del vector de deformaciones y tensiones

Para este ejercicio se logró entrenar una relación constitutiva elástica lineal con datos de deformaciones y tensiones que fueron obtenidos con deformaciones en un rango entre  $-0,0064$  y  $+0,0064$ . Se creó una matriz de datos con 64 estados de deformación (patrones de entrada) con los cuales se entrenó la red. Las componentes de cada vector patrón son  $\varepsilon_{11}$ ,  $\varepsilon_{22}$  y  $\gamma_{12}$ . Cada patrón se obtuvo evaluando  $E = 2,1e+6$  y  $\nu = 0,3$  en la Ecuación 11. La relación que se debería encontrar luego de entrenar la red es:

$$C = \begin{bmatrix} 2,8269e+06 & 1,2115e+06 & 0 \\ 1,2115e+06 & 2,8269e+06 & 0 \\ 0 & 0 & 8,0769e+05 \\ 1,2115e+06 & 1,2115e+06 & 0 \end{bmatrix} \quad (18)$$

La topología de la red es similar a la presentada en la figura 2, donde se posee un Perceptrón simple con 4 entradas y 4 salidas. La cuarta entrada es un valor fijo igual a 1 (umbral).

La labor del Perceptrón implementado es, entonces, lograr hallar las componentes de la relación constitutiva matricial, es decir, los pesos sinápticos de la red. Las entradas fueron escaladas a un rango entre  $-1$  y  $1$  al igual que las salidas. Luego, el resultado de la matriz de pesos es escalada para poder obtener

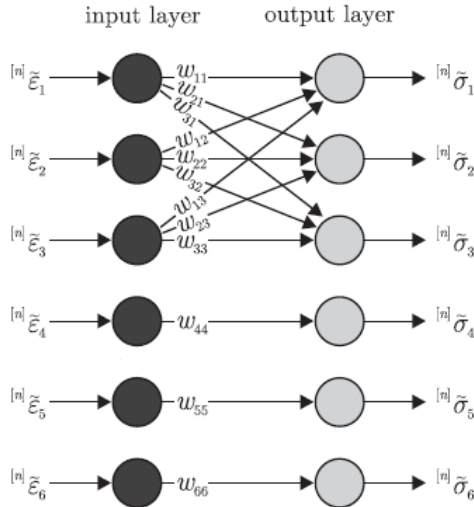


Figura 2: Red alimentada hacia adelante para modelo de material elástico lineal. Tomada de [5].

las componentes deseadas. Para ello debemos tener en cuenta la relación constitutiva a escala completa producto del aprendizaje  $\underline{C}_{nn}$ .

$$\underline{\sigma} = \underline{C}_{nn}\underline{\varepsilon} \quad (19)$$

Ahora tenemos en cuenta la relación a escala, es decir nuestra red neuronal, donde  $\underline{\sigma}'$  es el vector de salidas,  $\underline{\varepsilon}'$  es el vector de entradas y  $\underline{C}'_{nn} = \underline{W}$  es la matriz de pesos entre las dos capas.

$$\underline{\sigma}' = \underline{C}'_{nn}\underline{\varepsilon}' = \underline{W}\underline{\varepsilon}' \quad (20)$$

Siguiendo la convención para los subíndices de los pesos hallada en los textos, y apreciada en la figura 2 tenemos que

$$\underline{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \quad (21)$$

Ahora tenemos en cuenta las escalas de las entradas y las salidas

$$\underline{\sigma}' = A_\sigma \underline{\sigma} \quad y \quad \underline{\varepsilon}' = A_\varepsilon \underline{\varepsilon} \quad (22)$$

donde  $A_\sigma$  es el número que escala las tensiones y  $A_\varepsilon$  el que escala las deformaciones, ambas, a un rango entre -1 y 1. Reemplazando en la ecuación 20 y reordenando obtenemos

$$\underline{\sigma} = \left( \frac{A_\varepsilon}{A_\sigma} \underline{C}'_{nn} \right) \underline{\varepsilon} \quad (23)$$

Por igualdad con la ecuación 19 finalmente obtenemos la ecuación para escalar los pesos y transformarlos a componentes de la matriz constitutiva:

$$\underline{C}_{nn} = \left( \frac{A_\varepsilon}{A_\sigma} \underline{C}'_{nn} \right) = \left( \frac{A_\varepsilon}{A_\sigma} \underline{W} \right) \quad (24)$$

### 5.1.2. Función de activación lineal

Para lograr el entrenamiento exitoso de la red se empleó una red sin capas ocultas y una función de activación lineal para modelar el comportamiento de un material elástico lineal, tal como lo sugiere la referencia [5], esto es

$$g(h) = h \quad (25)$$

su derivada sería entonces

$$g'(h) = 1 \quad (26)$$

### 5.1.3. Escalamiento de datos de entrada y salida

El escalamiento de los datos de entrada y salida se realizó de acuerdo a lo propuesto por la ecuación 14. Las entradas fueron escaladas en un rango entre -1 y 1, al igual que las salidas. Para esto, todos los valores fueron divididos sobre el dato con mayor valor absoluto.

### 5.1.4. Estados enseñados a la red

En la figura 3 se aprecia una gráfica, donde cada punto representa un estado de tensiones y deformaciones. Debido a que un estado está representado por tres deformaciones y cuatro tensiones es necesario graficar las deformaciones y tensiones equivalentes. Las fórmulas para el caso más general (3D), son:

$$\begin{aligned} \epsilon_{eq} = \frac{\sqrt{2}}{3} [ & (\epsilon_x - \epsilon_y)^2 + (\epsilon_y - \epsilon_z)^2 + (\epsilon_z - \epsilon_x)^2 \\ & + 6(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)]^{1/2} \end{aligned} \quad (27)$$

$$\begin{aligned} \sigma_{eq} = \frac{1}{\sqrt{2}} [ & (\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 \\ & + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)]^{1/2} \end{aligned} \quad (28)$$

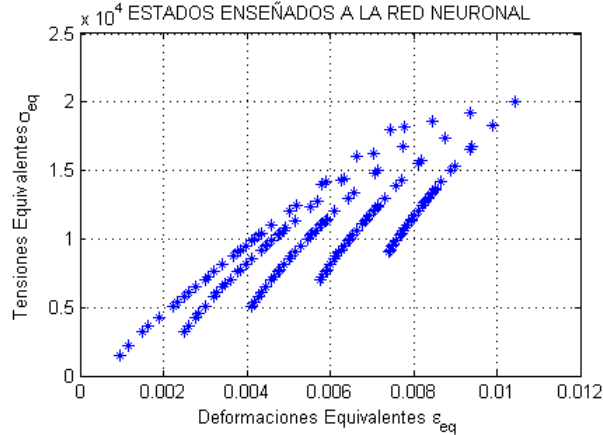


Figura 3: Estados enseñados a la red para modelo de material elástico lineal e isotrópico. Campo de estados para 1000 parejas de patrones enseñadas.

### 5.1.5. Resultados

En la tabla 1 se resumen los parámetros de simulación empleados para obtener la relación constitutiva de un material elástico, lineal e isotrópico a partir de datos de deformaciones (entradas) y tensiones (salidas). Al algoritmo de Backpropagation se le adicionó la teoría del momentum para el cálculo de los deltas de pesos, como se estudió en la sección 4.5.

| Parámetro                               | Valor                                   |
|---|---|
| Número de Patrones                      | 64                                      |
| Constante de aprendizaje $\eta$         | 0.001                                   |
| Constante de momentum $\alpha$          | 0.9                                     |
| Error RMS mínimo $E_{rms,min}$          | 1e-10                                   |
| Convergencia: Iteración, $E_{rms}$      | 60, 9.71e-011                           |
| Rango de deformaciones de entrenamiento | $-0,0064 \leq \varepsilon \leq +0,0064$ |

Cuadro 1: Parámetros de simulación para modelo elástico lineal isotrópico.

La relación constitutiva encontrada, luego de escalar los pesos de acuerdo a la ecuación 24 es:

$$C_{nm} = \begin{bmatrix} 2,8269e+06 & 1,2115e+06 & \rightarrow 0 & \rightarrow 0 \\ 1,2115e+06 & 2,8269e+06 & \rightarrow 0 & \rightarrow 0 \\ \rightarrow 0 & \rightarrow 0 & 8,0769e+05 & \rightarrow 0 \\ 1,2115e+06 & 1,2115e+06 & \rightarrow 0 & \rightarrow 0 \end{bmatrix} \quad (29)$$

$\rightarrow 0$  indica que los pesos en esas posiciones tienden a cero conforme se disminuye el  $E_{rms,min}$  permisible.

## 5.2. Material elástico lineal: RNA entrenada con componentes principales del modelo de material 2D

### 5.2.1. Entrenamiento sólo con componentes principales (aplicable a la realidad)

En la sección 5.1.1 se entrenó una red disponiendo de datos de deformaciones y tensiones normales y cortantes. En ésta sección, y para éste ejercicio, tomaremos sólo las componentes normales, debido a que los ensayos axiales reales no producen datos de deformación ni tensión cortantes por lo complejo de su medición. Para entrenar la red se utilizará la técnica de enriquecimiento de datos propuesta por Shin & Pande en la Referencia [12].

### 5.2.2. Sobre los ensayos axiales

Una posibilidad para la obtención de los datos con los cuales se podría entrenar la red proviene de ensayos axiales sobre una muestra del material bajo estudio, la cual es sometida a ciertas configuraciones de carga que garantizan la uniformidad del campo de tensiones y deformaciones. Una vez los datos de tensiones y deformaciones de los ensayos son capturados, se utilizan para entrenar la red que modela la relación constitutiva. Las mediciones habrán sido tomadas de mediciones sobre el volumen finito de la muestra, pero pueden ser empleadas para entrenar la relación aunque ésta esté definida para relacionar estados que están definidos en un sólo punto del material. Esto debido a los campos uniformes de tensiones y deformaciones que se logran durante los experimentos.

### 5.2.3. Datos de entrenamiento de la red

En lugar de usar datos experimentales reales para el material se usarán datos generados (también reciben el nombre de 'sintéticos') a partir de la ya conocida relación constitutiva matricial. Para éste propósito se emplearon parámetros típicos usados para modelar metales con una relación constitutiva isotrópica. Definimos el módulo de Young  $E = 2,1e6 Pa$  y el coeficiente de Poisson  $\nu = 0,3$ . Los patrones de tensión y deformación serán generados de acuerdo a lo sugerido por la Referencia [2], donde usaremos datos de tensiones y deformaciones para condiciones de carga a tensión y compresión para cada uno de los dos ejes principales y serán complementados con la técnica de enriquecimiento

de datos vista de la sección 5.2.4.

Queremos generar datos sin términos cortantes, tal como son obtenidos los datos experimentales de las pruebas triaxiales, y comprobar, utilizando la técnica de enriquecimiento de datos, que es posible obtener de nuevo la relación constitutiva elástica lineal e isotrópica. Los datos de deformaciones principales serán obtenidos mediante la siguiente relación:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = M \begin{bmatrix} 1 - \nu & \nu \\ \nu & 1 - \nu \\ \nu & \nu \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} \quad (30)$$

Donde  $\sigma_1$  y  $\sigma_2$  son las tensiones principales y  $\epsilon_1$  y  $\epsilon_2$  son las deformaciones principales. La única diferencia entre este ejercicio y el anterior son los datos de entrenamiento de la red. En la sección 5.2.4 se describe la técnica de enriquecimiento de datos, para aprovechar las componentes principales y generar con ellas un set de patrones con más información disponible para entrenar la red.

#### 5.2.4. Enriquecimiento de datos

Las parejas de vectores de tensiones y deformaciones que provienen de las pruebas triaxiales son en realidad tensiones y deformaciones principales. Esto significa que son mediciones tomadas a lo largo de ejes ortogonales entre sí, lo que en términos de campos de tensiones y deformaciones implica que no existen componentes cortantes. Si utilizáramos las componentes principales para entrenar la red ésta tendría que extrapolar los datos cortantes, debido a que no fue entrenada con parejas de vectores que contuvieran esta información. Esto conduciría a grandes imprecisiones en la respuesta tensión-deformación de la red.

Para superar esta limitante, Shin & Pande [12] propusieron una estrategia de enriquecimiento de datos para crear parejas de vectores (patrones) transformando tensiones y deformaciones a valores donde las componentes cortantes sean diferentes de cero. Para el caso bidimensional, la transformación de un vector de deformaciones principales rotado un ángulo  $\theta$  con respecto al eje  $X$  es como sigue:

$$\epsilon_x = C + D \cos(2\theta)$$

$$\epsilon_y = C - D \cos(2\theta)$$

$$\gamma_{xy} = 2D \sin(2\theta)$$

donde

$$C = (1/2)(\epsilon_1 + \epsilon_2)$$

$$D = (1/2)(\epsilon_1 - \epsilon_2)$$

Para las tensiones:

$$\sigma_x = A + B \cos(2\theta)$$

$$\sigma_y = A - B \cos(2\theta)$$

$$\tau_{xy} = B \sin(2\theta)$$

donde

$$A = (1/2)(\sigma_1 + \sigma_2)$$

$$B = (1/2)(\sigma_1 - \sigma_2)$$

Este método produce una gran cantidad de datos dependiendo del número de transformaciones elegidas para generarlos. Entre los datos expandidos hay muchas parejas de vectores repetidas de modo que se necesitan procesar previamente los datos para filtrarlas. Para el entrenamiento de la red se empleará un ángulo incremental  $\Delta\theta = 15^\circ$  para rotar los ejes de tensión-deformación desde  $-45^\circ$  hasta  $+45^\circ$ .

#### 5.2.5. Estados enseñados a la red

En la figura 4 se aprecia una gráfica, donde cada punto representa un estado de tensiones y deformaciones. Debido a que un estado está representado por tres deformaciones y cuatro tensiones es necesario graficar las deformaciones y tensiones equivalentes. Ver ecuaciones 27 y 28. Los puntos graficados en la figura 4 corresponden a estados de deformación y tensión luego del enriquecimiento de datos.

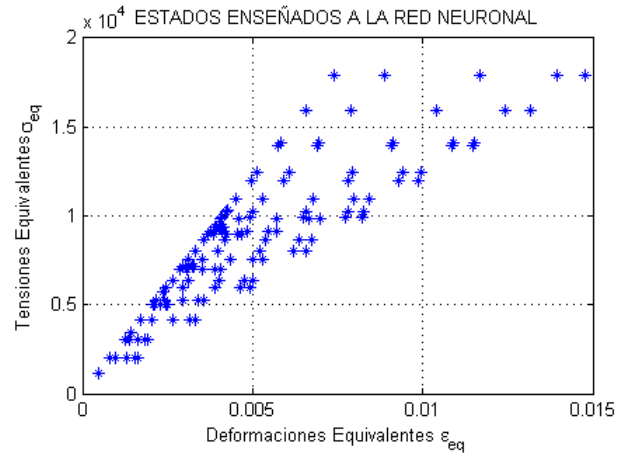


Figura 4: Campo de estados para 730 parejas de patrones.

### 5.2.6. Resultados

En la tabla 3 se resumen los parámetros de simulación empleados para obtener la relación constitutiva elástica lineal de este ejercicio. Los resultados fueron obtenidos entrenando sólo con 76 patrones, los cuales fueron obtenidos luego del enriquecimiento y del filtrado de los patrones repetidos, y partiendo de tan solo 16 parejas de vectores de deformaciones y tensiones principales.

| Parámetro                               | Valor                                   |
|---|---|
| Número de Patrones enseñados            | 76                                      |
| Constante de aprendizaje $\eta$         | 0.001                                   |
| Constante de momentum $\alpha$          | 0.9                                     |
| Error RMS mínimo $E_{rms,min}$          | 1e-10                                   |
| Convergencia: Iteración, $E_{rms}$      | 314, 9.50e-011                          |
| Rango de deformaciones de entrenamiento | $-0,0064 \leq \varepsilon \leq +0,0064$ |

Cuadro 2: Parámetros de simulación para modelo elástico lineal isótropo.

La relación constitutiva encontrada, luego de escalar los pesos de acuerdo a la ecuación 24 es:

$$C_{nm} = \begin{bmatrix} 2,8269e+06 & 1,2115e+06 & \rightarrow 0 & \rightarrow 0 \\ 1,2115e+06 & 2,8269e+06 & \rightarrow 0 & \rightarrow 0 \\ \rightarrow 0 & \rightarrow 0 & 8,0769e+05 & \rightarrow 0 \\ 1,2115e+06 & 1,2115e+06 & \rightarrow 0 & \rightarrow 0 \end{bmatrix} \quad (31)$$

$\rightarrow 0$  indica que los pesos en esas posiciones tienden a cero conforme se disminuye el  $E_{rms,min}$  permisible.

### 5.3. Material elasto-plástico no lineal: RNA entrenada con componentes principales

El objetivo de éste ejercicio es entrenar un perceptrón de dos capas ocultas a partir de deformaciones y tensiones obtenidas de datos sintéticos a partir de un programa de Matlab que calcula deformaciones y tensiones de un elemento finito sometido a cargas, ver figura 5. El material considerado es no lineal debido a que sufre deformación plástica cuando la tensión de Von Mises supera la tensión de fluencia. Cuando esto sucede, el material sufre endurecimiento por deformación, lo que hace depender el estado actual del historial de deformaciones y tensiones en ese punto. Para modelar este tipo de problemas numéricamente

se requiere de métodos iterativos para satisfacer las ecuaciones del modelo en cada paso de aplicación de la carga.

#### 5.3.1. Entrenamiento con componentes principales (aplicable a la realidad)

El objetivo más importante de este ejercicio es obtener unos pesos que permitan estimar tensiones a partir de deformaciones, teniendo como datos de entrada y salida sólo deformaciones y tensiones principales, que pueden ser obtenidas de ensayos reales. Se logrará el objetivo si se logran pesos que permitan buenas estimaciones al someter al elemento finito mencionado a configuraciones de carga diferentes a las configuraciones con las cuales se entrenó la red.

#### 5.3.2. Datos de entrenamiento de la red

Los pasos para la obtención de los datos de entrenamiento es como sigue:

1. Se somete un elemento finito para modelar deformación plana elasto-plástica de 4 nodos a configuraciones incrementales de carga a tensión, combinando 10 incrementos de carga horizontales con 10 incrementos verticales. En la figura 5 se aprecian tres estados principales de carga para el entrenamiento.
2. Se calculan deformaciones y tensiones principales para cada combinación de cargas. Esto dará 3 deformaciones y 4 tensiones, igual que para los ejercicios 1 y 2 estudiados anteriormente.
3. Se realiza un enriquecimiento de datos, como sugieren Drakos & Pande en la referencia [2].
4. El punto anterior generará algunos patrones repetidos, los cuales son eliminados.
5. Se escalan los datos de entrada y salida, como se vió en la sección 4.2.

Los estados de deformaciones y tensiones están compuestos por 3 y 4 componentes, respectivamente. Para facilitar la visualización de dichos estados los transformaremos a deformaciones y tensiones equivalentes, igual que como vimos en la sección 5.1.4.

#### 5.3.3. Curva de carga: material elasto-plástico bilineal

En la figura 6 se aprecian los estados de entrenamiento sin enriquecimiento, es decir, aún no aptos para emplearse en el entrenamiento. En ella se aprecia una zona elástica, antes para  $0 \leq \sigma_{eq} \leq 2500$  y



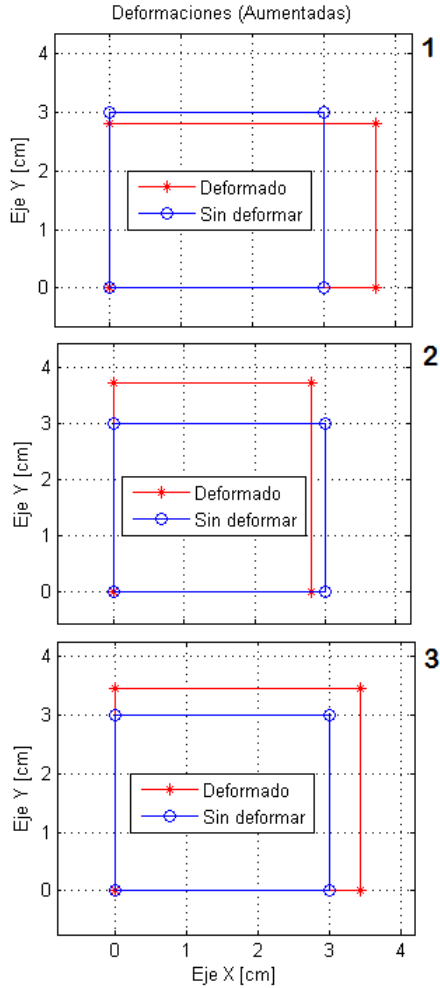


Figura 5: 3 de 120 estados de carga para entrenamiento de RNA. Los puntos dentro del elemento poseen estados de deformación y tensión uniformes.

una plástica para  $2500 \leq \sigma_{eq} \leq \infty$ . El endurecimiento por deformación se produce para estados dentro de la zona plástica.

### 5.3.4. Enriquecimiento de datos

Esta generación adicional de datos es importante, ya que es la que permitirá poder estimar las tensiones de corte cuando se posean deformaciones con componentes cortantes. Este enriquecimiento aumenta los patrones de entrenamiento de 120 a 676, y se realiza de igual forma que para el Ejercicio 2 en la sección 5.2.4.

En la figura 7 se observan los estados de entrenamiento de la red después de haber efectuado el enriquecimiento de datos.

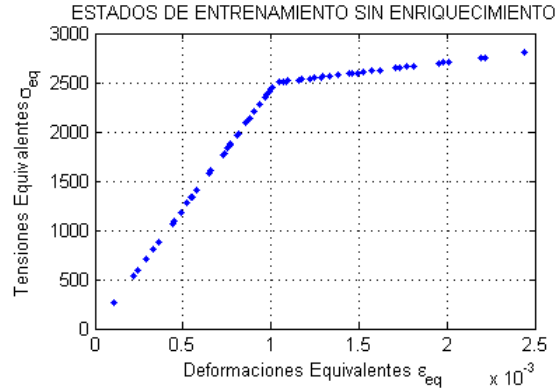


Figura 6: Estados de entrenamiento antes del enriquecimiento para 120 parejas de patrones. Material elasto-plástico bajo varias configuraciones de carga.

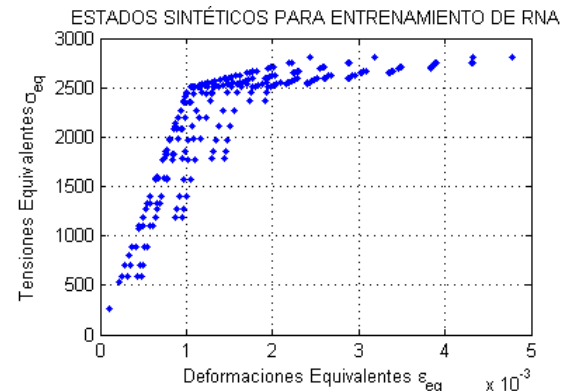


Figura 7: Estados de entrenamiento después del enriquecimiento para 676 parejas de patrones.

### 5.3.5. Topología de la red

Luego de numerosos intentos por obtener el mejor aprendizaje posible, se pudo identificar que la topología [4-12-8-4] es la que da mejores resultados por aproximar en menor tiempo y con mayor exactitud los patrones deseados, con 4 neuronas de entrada (1 es bias) y 4 neuronas de salida, y con 2 capas ocultas de 12 y 8 neuronas cada una. La selección de ésta topología sigue aproximadamente la configuración que muestran Drakos & Pande en la página 652 de la referencia [2].

### 5.3.6. Resultado del entrenamiento

El objetivo es lograr estados aproximados por la red lo más similares posibles a los estados de entrenamiento, inicialmente. Para este ejercicio se disminuyó la constante de aprendizaje  $\eta$  cada 1500 itera-

ciones (epochs), partiendo de  $\eta = 0,05$  y disminuyéndola por 0.75 cada vez. Se utilizó la adición del términos de momentum para el cálculo de  $\Delta w_{ij}$  visto en la sección 4.5. En la tabla 3 se halla un resumen de los parámetros empleados, mientras que en la figura 8 se aprecian los estados logrados por la red luego del proceso de aprendizaje por Backpropagation (en rojo).

| Parámetro                          | Valor           |
|------------------------------------|-----------------|
| Número de Patrones enseñados       | 676             |
| Constante de aprendizaje           | $\eta < 0,05$   |
| Constante de momentum $\alpha$     | 0.9             |
| Convergencia: Iteración, $E_{rms}$ | 30000, 4.35e-03 |

Cuadro 3: Parámetros de simulación para modelo elástico lineal isótropo.

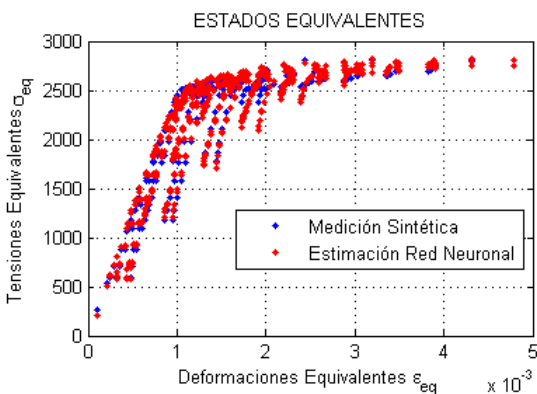


Figura 8: Superposición de estados de entrenamiento y estados aproximados por la red luego del aprendizaje.

## 6. Pruebas

Se requieren pruebas para constatar que el modelo constitutivo aprendido por la RNA es útil para estimar otros estados de tensión diferentes a los usados para el entrenamiento. Para ello se emplearán datos obtenidos de someter al elemento finito para deformación plana a configuraciones de carga diferentes a las usadas para obtener los datos de entrenamiento.

### 6.1. Prueba con estados de deformación 1

Para este caso las cargas aplicadas son dos verticales y dos horizontales, una en cada nodo con incrementos iguales para lograr un estado uniforme de deformaciones. La configuración de cargas puede verse en la figura 9. El estado de deformación final

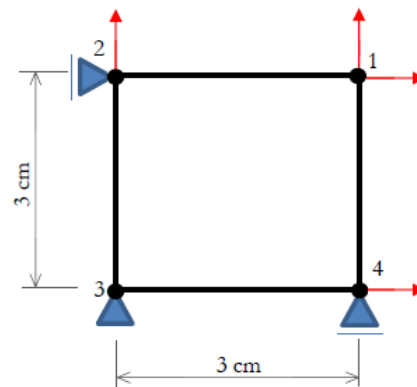


Figura 9: Configuración de carga 1. Rango de 500 a 9500 kgf, incrementos de 1000 kgf.

del elemento es similar al presentado en la figura 5.

En la figura 10 observamos la primera comprobación del funcionamiento del modelo constitutivo obtenido con la RNA. Los patrones de deformaciones usados como entrada son desconocidos para la red.

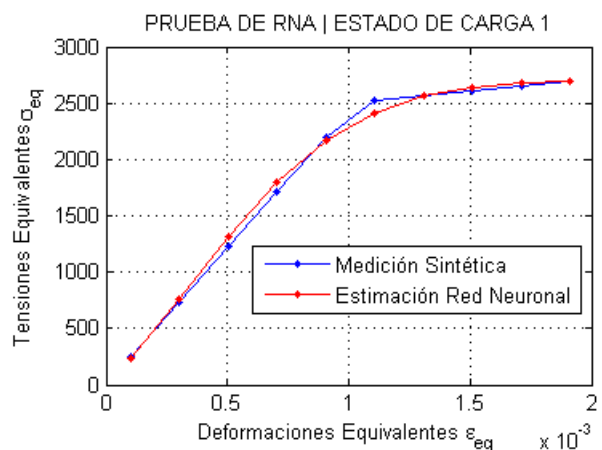


Figura 10: Estados para configuraciones de carga 1. Estado uniforme de deformaciones dentro del elemento.

## 6.2. Prueba con estados de deformación 2

Las cargas aplicadas son dos verticales y una horizontal con incrementos iguales. Por su asimetría los estados de deformación dentro del elemento no son uniformes (iguales), por lo tanto existen diferentes estados en cada punto de integración (son 4 puntos de Gauss, dispuestos de forma similar a los nodos, pero dentro del elemento). La configuración de cargas puede verse en la figura 11.

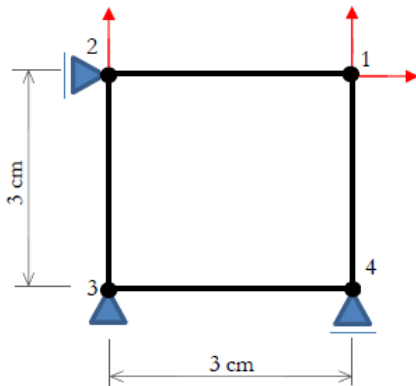


Figura 11: Configuración de carga 2. Rango de 500 a 9500 kgf, incrementos de 1000 kgf.

El estado de deformaciones del elemento se observa en la figura 12.

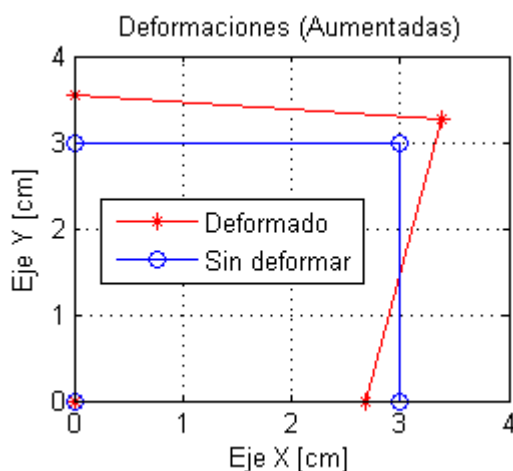


Figura 12: Gráfico de deformaciones del elemento para Caso de prueba 2.

En la figura 13 observamos la segunda comprobación del modelo de material elastoplástico. La curva azul identifica los estados sintéticos obtenidos por el

que podría ser un probeta real, mientras que la curva roja los estados predichos por la RNA.

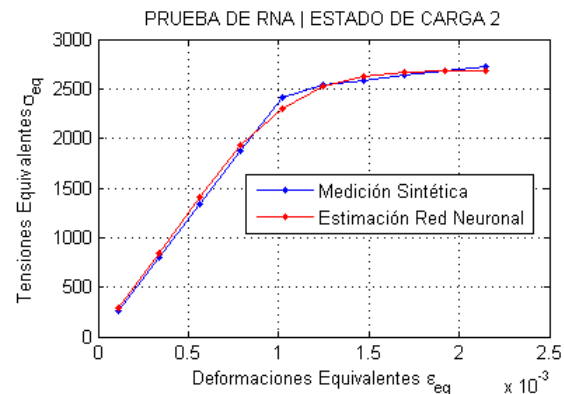


Figura 13: Estados para configuraciones de carga 2. Estado no-uniforme de deformaciones dentro del elemento. Datos para punto de Gauss 1 (abajo-izquierda).

## 6.3. Prueba con estados de deformación 3

La configuración de cargas para éste caso de prueba es la que vemos en la figura 14.

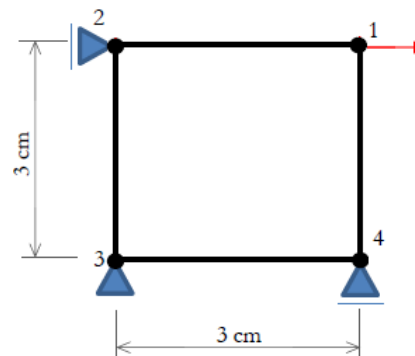


Figura 14: Configuración de carga 3. Rango de 0 a 20000 kgf, incrementos de 2000 kgf.

Las deformaciones del elemento plano sometido a las cargas de la figura 14 están ilustradas en la figura 15. Finalmente, en la figura 16 hallamos la tercera comprobación. Las cargas aplicadas son dos horizontales en los nodos 1 y 2, para lograr un efecto predominante de corte. Éste grupo de configuraciones de carga es probablemente la más alejada de aquellas con las cuales se entrenó la red.

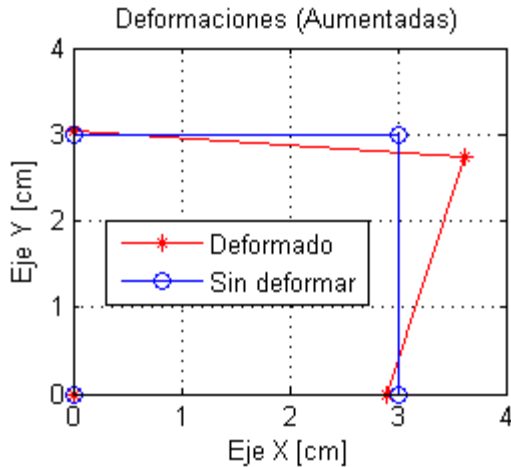


Figura 15: Gráfico de deformaciones del elemento para Caso de prueba 3.

## 7. Comentarios y Conclusiones

- Caso de prueba 1: si comparamos los números del modelo de material  $C_{nn}$  con los números del modelo original  $C$  de la ecuación 18 podemos constatar que el aprendizaje de la relación constitutiva elástica lineal isótropa a partir de datos de tensiones y deformaciones se efectuó de forma exitosa ya que se logró estimar los valores de las componentes de la matriz sólo a partir de datos de deformaciones y tensiones. Como otro aporte que puede resultar interesante se destaca que obtener la matriz constitutiva permite también hallar las dos constantes del material, por ahora, sólo para el modelo elástico lineal. Las constantes que se puede hallar son el módulo elástico o de Young  $E$  y el coeficiente de Poisson  $\nu$ .
- Caso de prueba 2: Se logró obtener la relación constitutiva a partir de parejas de patrones de deformación y tensión principales. Esto demuestra que a partir de datos de pruebas triaxiales reales y utilizando la técnica de enriquecimiento de datos usada por Shin & Pande en la referencia [12] es posible obtener las componentes de la matriz constitutiva para emplearse en modelos que emplean materiales elásticos lineales e isótropos.
- Las tres pruebas de la RNA sugieren que ante estados de deformación desconocidos para la red, es posible obtener estados de tensión aproximados que describen el material utilizado para extraer los datos de entrenamiento. Esto indica que a partir de datos de deformaciones y ten-

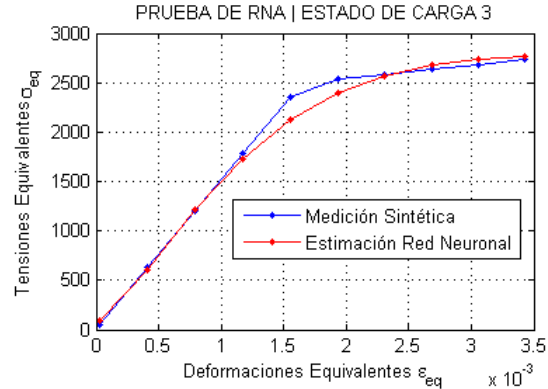


Figura 16: Estados para configuraciones de carga 3. Estado no-uniforme de deformaciones dentro del elemento. Datos para punto de Gauss 1 (abajo-izquierda).

siones de un material con comportamiento no-lineal también podemos obtener una estimación del modelo si empleamos redes de múltiples capas cuyos pesos puedan ajustarse al comportamiento no lineal del material. En las pruebas, la zona donde peores aproximaciones se obtienen es aquella donde se da el cambio de zona de elástico a plástico, donde la función de activación no puede aproximar fielmente el cambio discontinuo de una recta a otra.

- La duración de los ensayos numéricos iterativos para entrenar el perceptrón y lograr un error aceptable entre la salida deseada y la salida actual de la red puede ser bastante prolongada en aquellos casos donde la cantidad de patrones a enseñar son numerosos. Una alternativa para optimizar los tiempos de aprendizaje es paralelizar las partes del código que permitan ser paralelizadas, como los productos escalares de las multiplicaciones de matriz por vector, donde repartiríamos el mismo vector a varios procesos y luego dividiríamos las matrices y las repartiríamos por partes a cada proceso. Los resultados se retornarían después al proceso principal. De lograr un código escalable obtendríamos un ahorro en tiempo muy significativo.
- Una implementación para futuros trabajos es comprobar el comportamiento de la red para condiciones de borde diferentes a las estudiadas, donde se generarían otros estados de deformación que también deberían ser estudiados para validar el modelo de material con RNAs.
- El algoritmo original de back-propagation puede

ser muy lento. Se han propuesto variaciones en este algoritmo para conseguir mayores velocidades de aprendizaje. Los investigadores de la Referencia [7] utilizaron el algoritmo RPROP [11] para entrenar sus modelos debido a su confiabilidad mejorada y a la relativamente baja cantidad de parámetros de control.

## Referencias

- [1] CHEN, T., AND CHEN, H. Universal approximation to non-linear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Networks*, 6 (4) (1995), 911–917.
- [2] DRAKOS, S. I., AND PANDE, G. N. A neural network equivalent of hardening soil model of plaxis. *Numerical Methods in Geotechnical Engineering* (2006), 651–656.
- [3] DVORKIN, E., AND GOLDSCHMIT, M. *Nonlinear continua*. Springer, May 2005.
- [4] FLOOD, I., AND KARTAM, N. Neural networks in civil engineering. i: principles and understanding. *Journal of Computing in Civil Engineering*, 8(2) (1994), 131–147.
- [5] FREITAG, S., GRAF, W., AND KALISKE, M. A material description based on recurrent neural networks for fuzzy data and its application within the finite element method. *Computers and Structures*, 124 (2013), 29–37.
- [6] GHABOUSSI, J., PECKNOLD, D., ZHANG, M., AND HAJ-ALI, R. Autoprogressive training of neural network constitutive models. *Int. J. Numer. Meth. Engng*, 42 (1998), 105–126.
- [7] HASHASH, Y. M. A., JUNG, S., AND GHABOUSSI, J. Numerical implementation of a neural network based material model in finite element analysis. *Int. J. Numer. Meth. Engng*, 59 (2004), 989–1005.
- [8] HERTZ, J., KROGH, A., AND PALMER, R. G. *Introduction to the theory of neural computation*, vol. 1. Addison-Wesley, 1990.
- [9] LEFIK, M., BOSO, D., AND SCHREFLER, B. Artificial neural networks in numerical modelling of composites. *Computational Methods in Applied Mechanics Engineering*, 198 (2009), 1785–1804.
- [10] LIN, Y., AND CHEN, X.-M. A critical review of experimental results and constitutive descriptions for metals and alloys in hot working. *Materials and design*, 32 (2011), 1733–1759.
- [11] RIEDMILLER, M., AND BRAUN, H. A. A direct adaptive method for faster backpropagation learning: The rprop algorithm. *IEEE International Conference in Neural Networks, IEEE: San Francisco, New York* (1993), 586–591.
- [12] SHIN, H. S., AND PANDE. Enhancement of data for training neural network based constitutive models for geomaterials. *Proc. of the Eighth Intl. Symp. on Numerical Models in Geomechanics (NUMOG VIII), Rome, Italy, April* (2002), 141–146.
- [13] SUN, Y., ZENG, W., ZHAO, Y., QI, Y., MA, X., AND HAN, Y. Development of constitutive relationship model of ti600 alloy using artificial neural network. *Computational Materials Science*, 48 (2010), 686–691.
- [14] SWINGLER, K. *Applying Neural Networks: A Practical Guide*. Morgan Kaufmann Publishers, California, 1996.