REDES NEURONALES 2024

Clase 21 parte 1

Lunes 4 de noviembre 2024

FAMAF, UNIVERSIDAD NACIONAL DE CÓRDOBA

INSTITUTO DE FÍSICA ENRIQUE GAVIOLA (UNC-CONICET)

J9/11/23 c22p2

Repasemos rápidamente lo visto hasta acá. Supongamos que tenemos un conjunto de datos que incluyen el resultado de una actividad inteligente de clasificación o regresión. Este supuesto conjunto viene dado por *p* entradas y sus correspondientes respuestas, las cuales fueron etiquetadas por algún o alguna experta

$$\mathcal{D} = (\mathbf{x}, \mathbf{y}) = \left\{ (\bar{x}^\mu, \bar{y}^\mu) \,/\, \bar{y}^\mu = f(\bar{x}^\mu), \mu = 1, 2, \dots, p \right\}$$

$$\bar{x}^\mu \in \mathbb{R}^N$$

$$\bar{y}^\mu \in \mathbb{R}^M$$

$$f : \mathbb{R}^N \longrightarrow \mathbb{R}^M$$

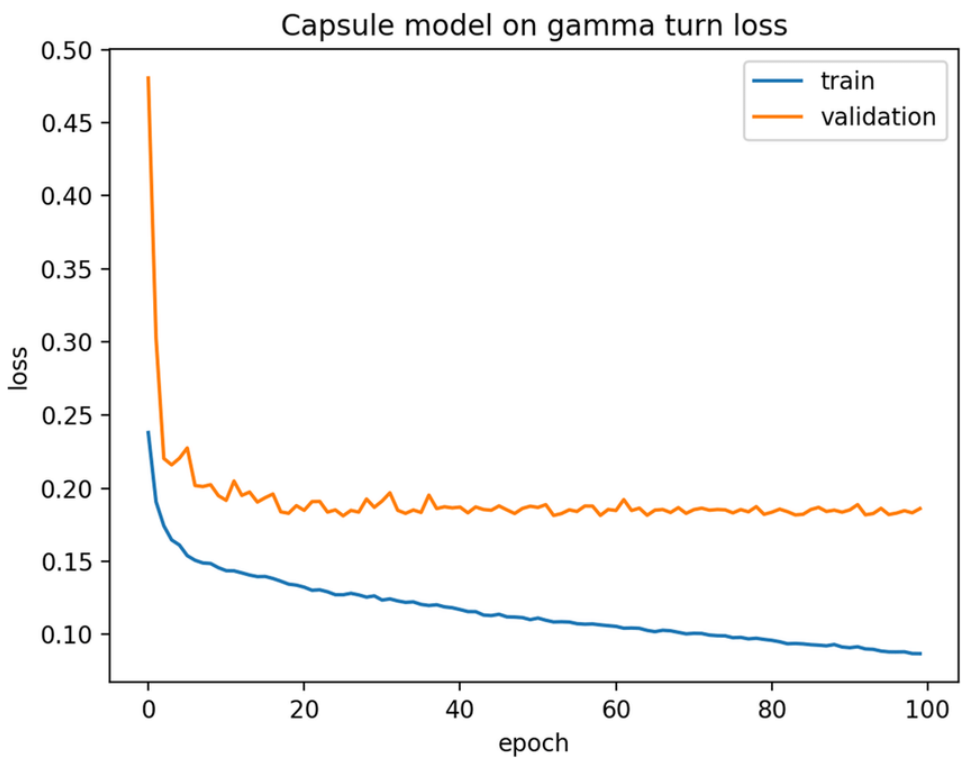
El conjunto de datos etiquetados nos dice que nuestro problema de aprendizaje supervisado requiere de *N* entradas y *M* salidas. Queremos construir una red neuronal feed-forward capaz de hacer inferencias sobre lo que hace la función objetivo *f* con los datos de entrada para generar la salida. Para eso la aproximaremos.

Aún cuando el problema sea de clasificación, a partir de ahora vamos a tener neuronas de salida con funciones de activación no lineales. Ya veremos cómo discretizar el problema para clasificar. Debemos definir el número de capas ocultas, el número de neuronas en cada capa oculta y la razón de aprendizaje η para lo cual deberemos hacer muchas pruebas.

Recordemos por favor que la red no nos dará la forma analítica de *f*, pero nos hará el favor de aproximar a *f* por una combinación de composiciones de sumas de funciones no lineales.

Dividiremos el conjunto de entrenamiento en tres subconjuntos:

- Conjunto de entrenamiento
- Conjunto de testeo
- Conjunto de verificación



¿PORQUÉ ES DIFÍCIL ENTRENAR UNA RED PROFUNDA?

Ahora que tenemos una descripción más profunda del mecanismo de entrenamiento y de aprendizaje en una red neuronal feed-forward con un número arbitrario de capas, podremos avanzar hacia lograr nuestro objetivo: construir redes neuronales muy profundas, o sea, con muchas capas ocultas, capaces de aprender a resolver problemas muy complejos a partir de grandes conjuntos de entrenamiento y con muchas sinapsis y umbrales.

La función loss o error es muy complicada

La función loss o error, que hasta aquí es el error cuadrático medio, es extremadamente compleja, por varias razones:

- Depende un número muy grande de parámetros, las sinapsis y los umbrales. En verdad esto no alcanza para tener complejidad. Podemos decir que este ingrediente aporta a tener un problema complicado.
- Las funciones de salida, son muchas concatenaciones de funciones fuertemente no lineales.

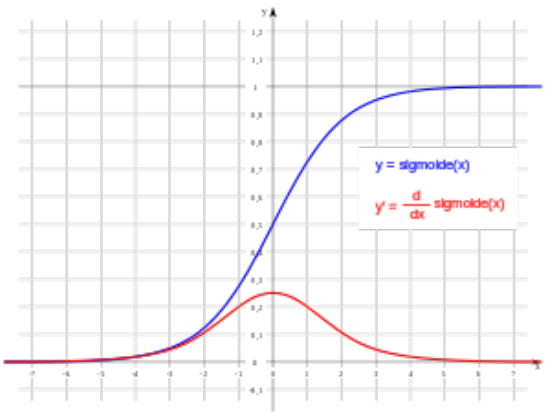
$$O_i = g\left(\sum_l w_l^{(L)}\, g\left(\sum_j w_j^{(L-1)} g\left(\cdots \sum_k w_k^{(1)} \xi_k\right)\right)\right)$$

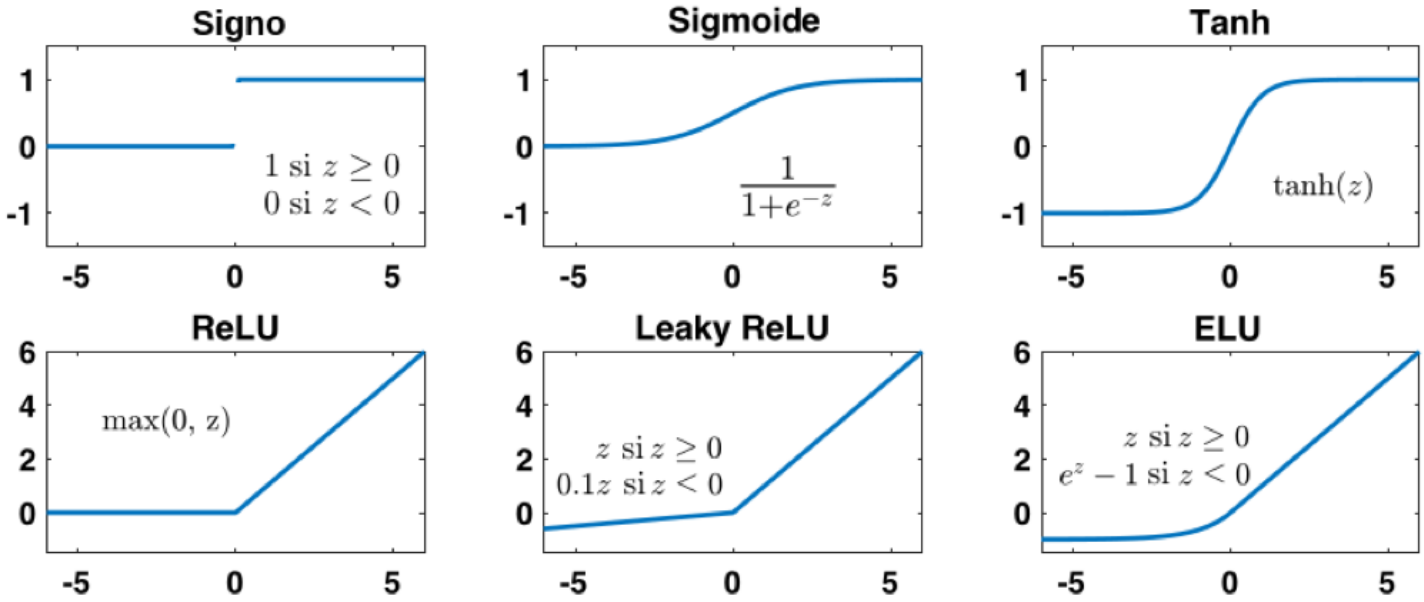
- Los parámetros (sinapsis más umbrales) pueden tomar tanto valores positivos como negativos. Luego,

LA FUNCIÓN LOSS ES ESTREMADAMENTE RUGOSA

- El descenso por el gradiente es determinista

- Las funciones sigmodes tiene casi siempre derivada pequeña



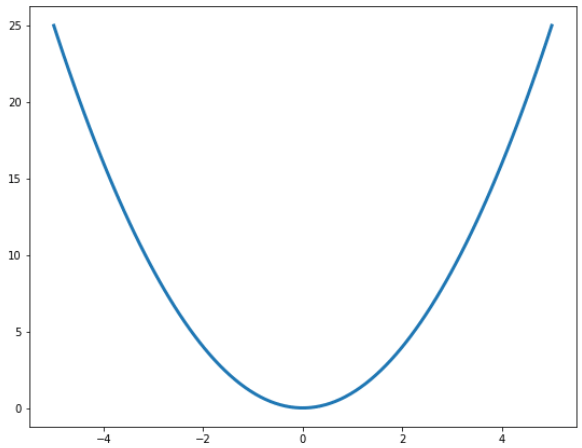

Signo
1
0
-1
-5
0
5
$1 \text{ si } z \geq 0$
$0 \text{ si } z < 0$
Sigmoide
$\frac{1}{1+e^{-z}}$
Tanh
$\tanh(z)$
ReLU
6
4
2
$\max(0, z)$
Leaky ReLU
$z \text{ si } z \geq 0$
$0.1z \text{ si } z < 0$
ELU
$z \text{ si } z \geq 0$
$e^z - 1 \text{ si } z < 0$

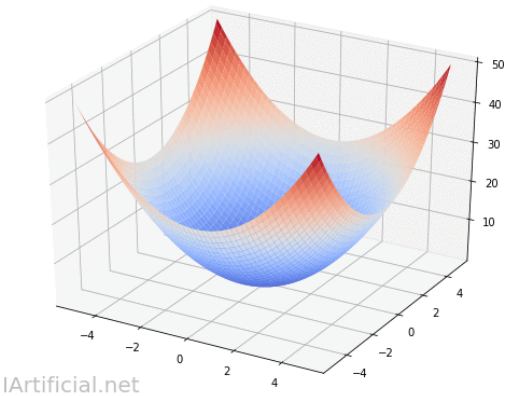
- Partimos muy lejos de la solución que buscamos

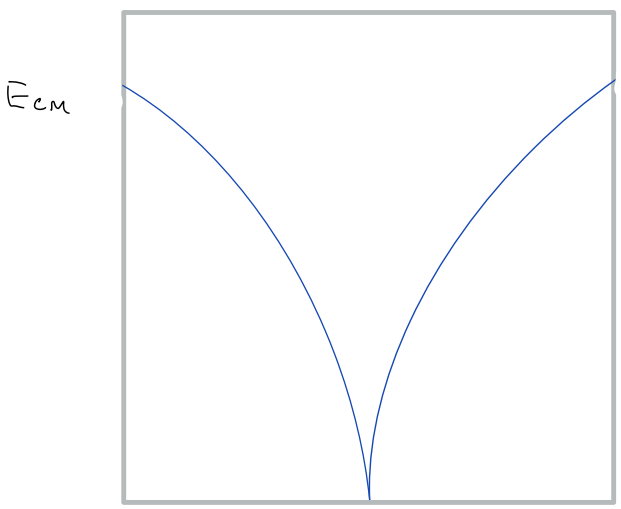
  Dada la alta dimensionalidad del espacio de los parámetros (sinapsis y umbrales), es muy muy probable que nuestra elección inicial de los parámetros caiga muy muy lejos del mejor mínimo o incluso de un buen

- La función ECM tiene mala concavidad

  La función error cuadrático medio tiene el problema de que lejos de un mínimo, tiene gradiente mayor que cerca del mínimo. Esto quiere decir que el método de descenso por el gradiente hacia un mínimo es rápido lejos del mínimo y lento cerca del mínimo. Nosotros en cambio aprendemos lento lejos de la solución y rápido cerca.







A partir de ahora introduciremos las modificaciones necesarias para superar todas estas limitaciones.

Es muy difícil definir el número de parametros

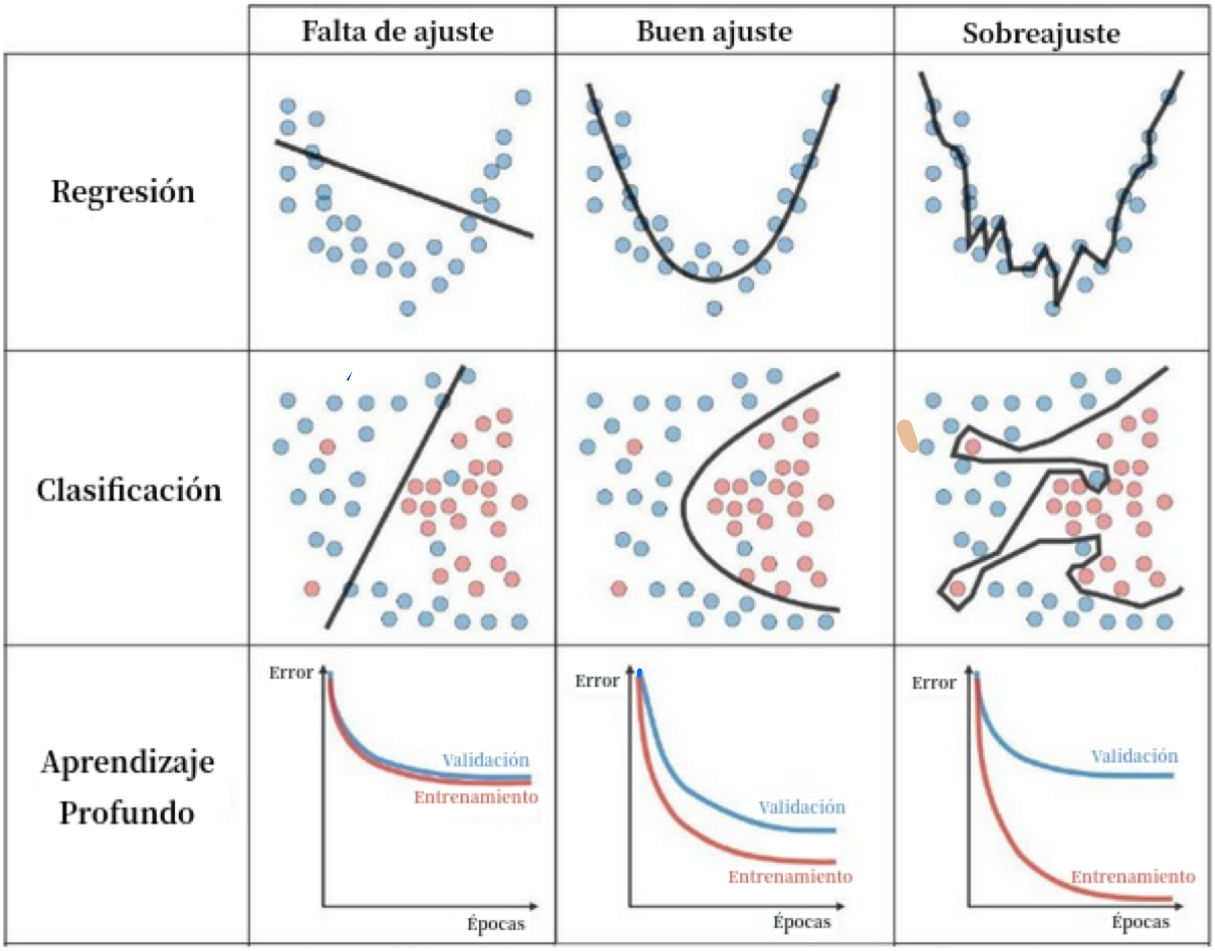


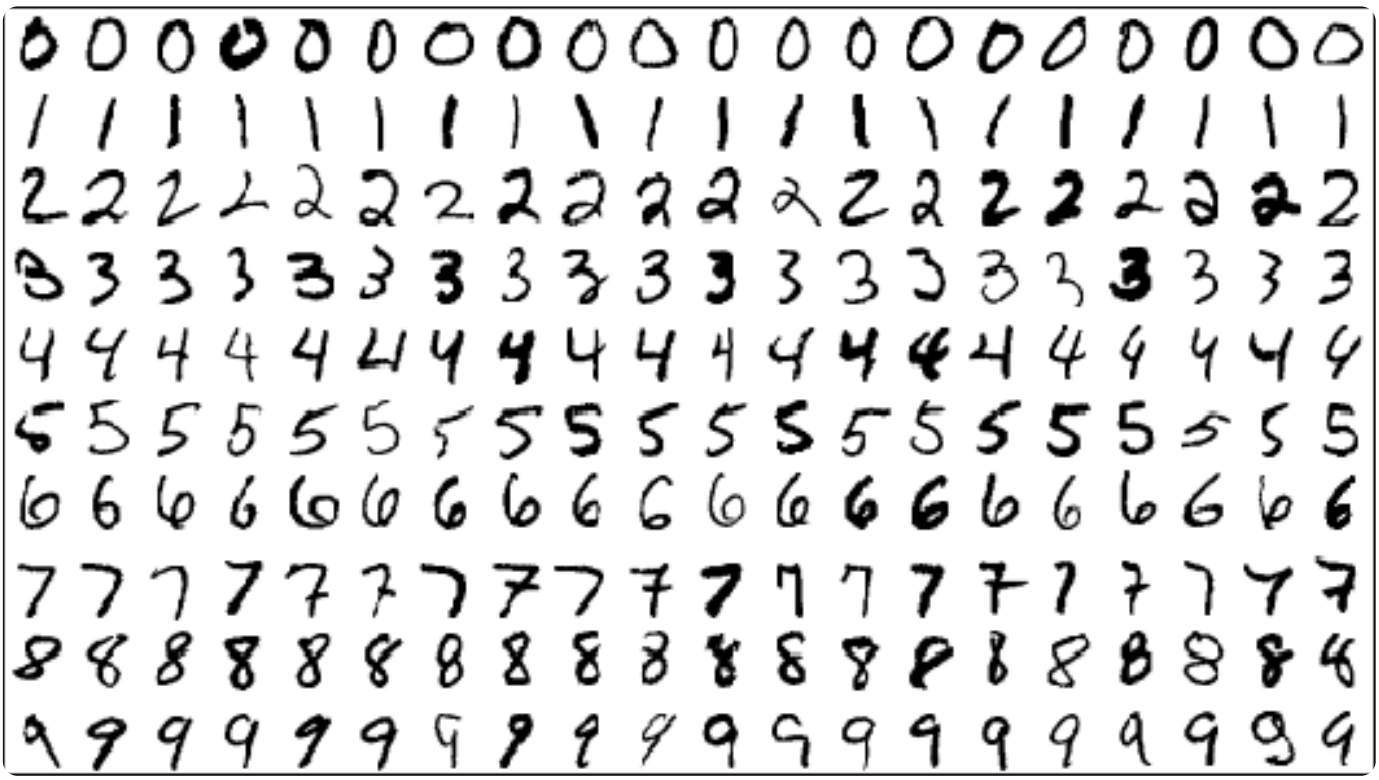
Figura 2.2. Comparación de un buen ajuste contra la falta de ajuste o sobreajuste.[2]

LA SUPRESIÓN DEL GRADIENTE

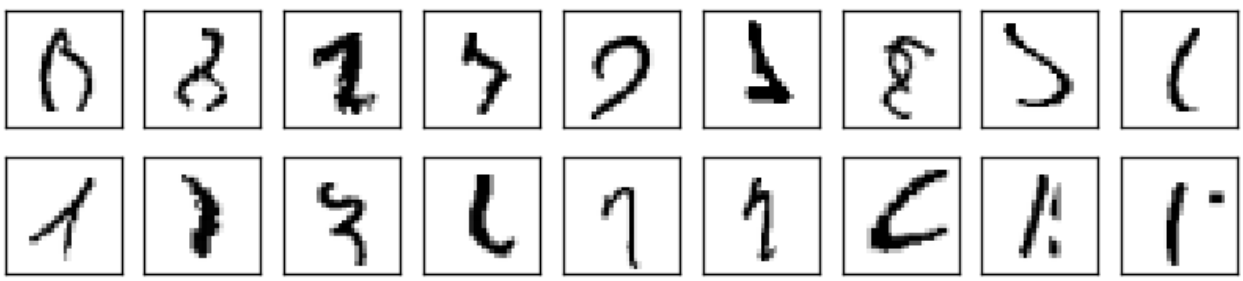
Vamos a analizar un experimento que ilustra este fenómeno. Armamos una red neuronal feed-forward para reconocer dígitos escritos a mano. Para ello consideramos un data set muy famoso llamado MNIST. El nombre viene de Modified National Institute for Standars and Technology dataste. El National Institute for Standars and Technology es equivalente a nuestro INTI en los Estados Unidos de Norteamérica. Son 70000 imágenes de dígitos escritos a mano y luego digitalizados.

Todas las imágenes digitalizadas fueron normalizadas en una caja cuadrada de 28x28 píxeles y en tonos de gris. Las 70000 imágenes surgieron a partir de la,digitalización de dígitos escritos a mano por:

- estudiantes voluntarios de secundario de EEUU y
- empleados voluntarios del Servicio de Correo de EEUU.



Algunos dígitos son difíciles de reconocer pero en el data set están las imágenes y lo que el voluntario dice que quiso escribir:

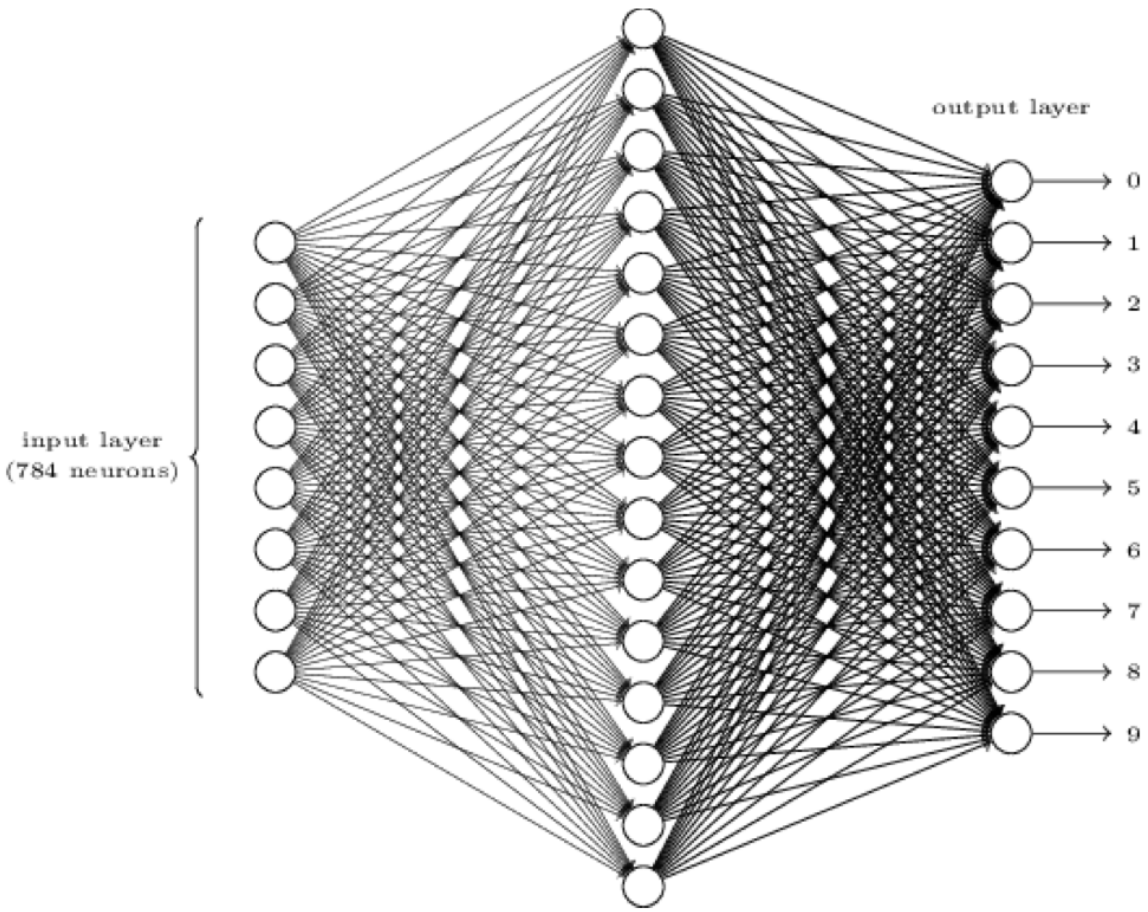


La naturaleza del conjunto de datos y del problema que queremos solucionar determinan la capa de entrada y el de salida:

- Entrada: es un vector de 28x28=784 píxeles que pueden tomar valores enteros entre 0 y 255 según la intensidad del gris (0 es negro y 255 es blanco). El vector se genera como un proceso de aplastamiento, o sea, pegando secuencialmente las 28 filas de la imagen una al lado de la otra, de arriba para abajo.

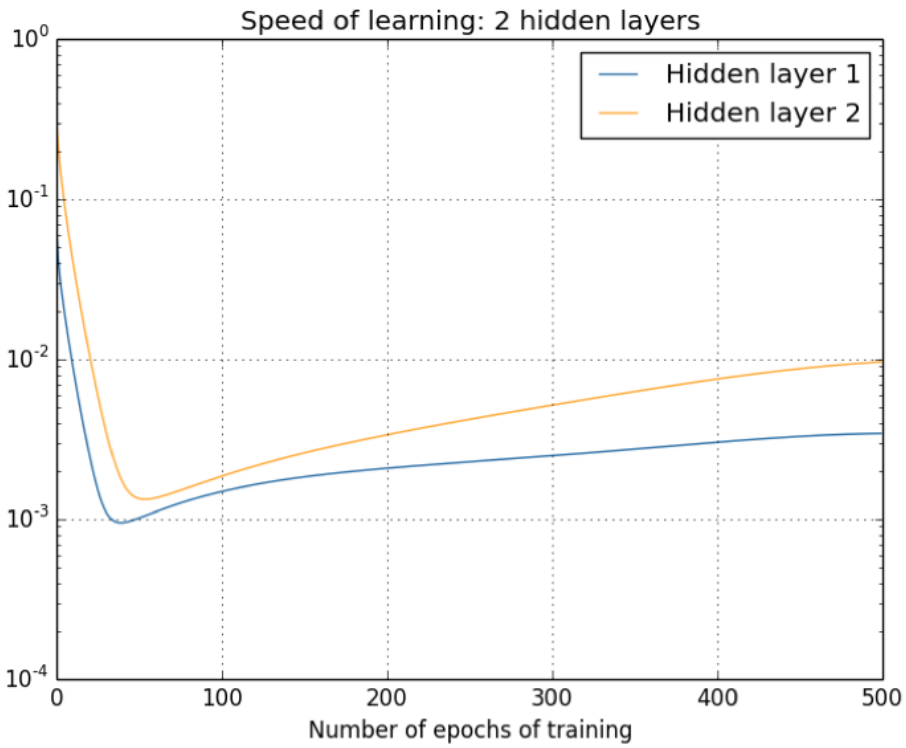
- Salida: 10 neuronas sigmoides (entre 0 y 1), una para que se active con cada uno de los diez tipos de dígitos. La categoría reconocida es aquella que corresponde a la neurona que tiene la mayor salida.

Este es un esquema de nuestra red con una capa oculta.



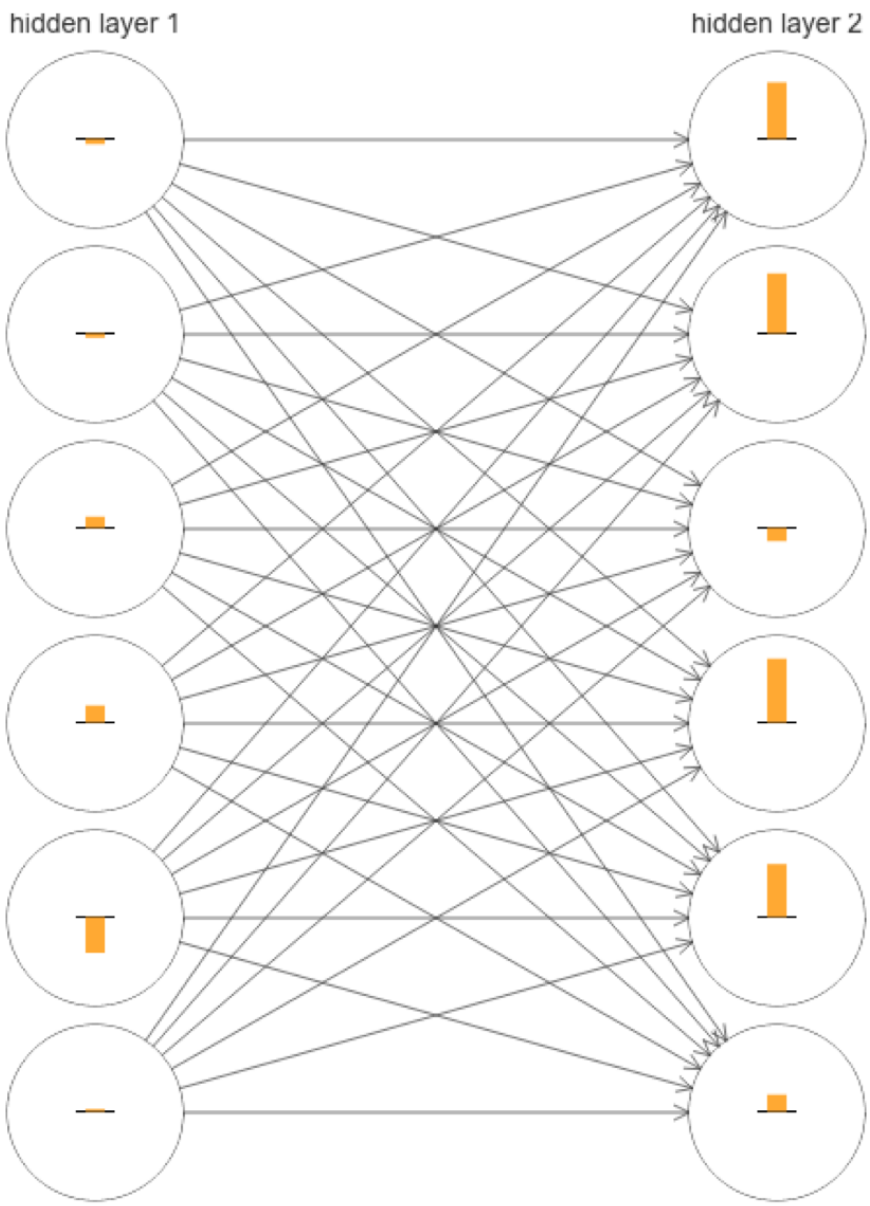
Este es un esquema de nuestra red con una capa oculta. Nuestro ejercicio será el siguiente: dado un número determinado de capas ocultas, las cuales irán aumentando, vamos a calcular el promedio de las componentes del gradiente en cada capa. No incluimos el valor de la razón de cambio (no es un promedio de los incrementos).

Comenzamos con dos capas ocultas de 100 neuronas cada una. En la figura siguiente les muestro dos curvas en las cuales se representa en el eje de ordenadas el promedio de los gradientes de los umbrales en cada capa y en una dada época. La épocas se representan en las abscisas.



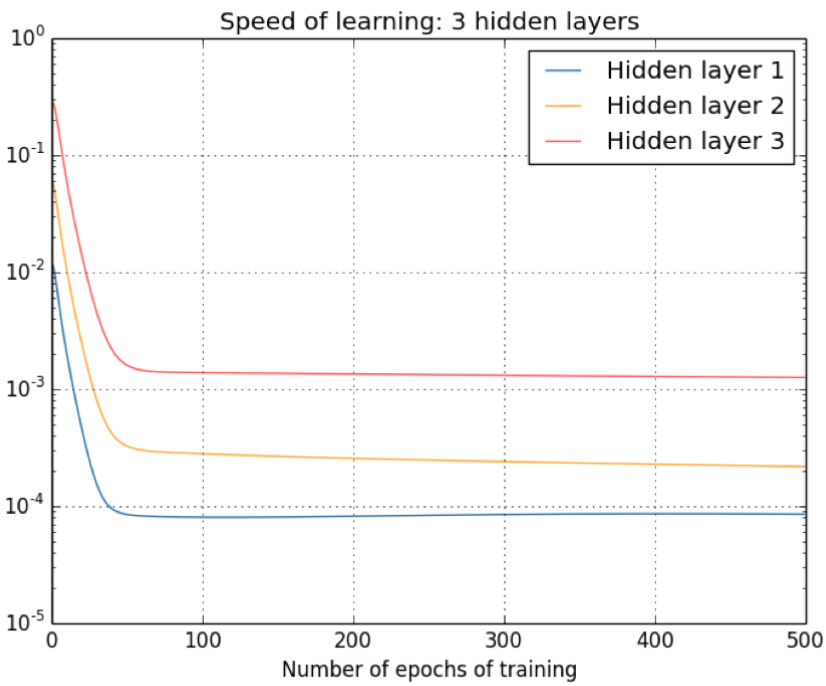
La curva amarilla corresponde al promedio de los módulos de las componentes del gradiente de los umbrales de la segunda capa oculta. La curva celeste corresponde al promedio de las componentes del gradiente de los umbrales de la primera capa oculta.

El mismo efecto se ve con los promedios de las sinapsis entre capas.


hidden layer 1
hidden layer 2

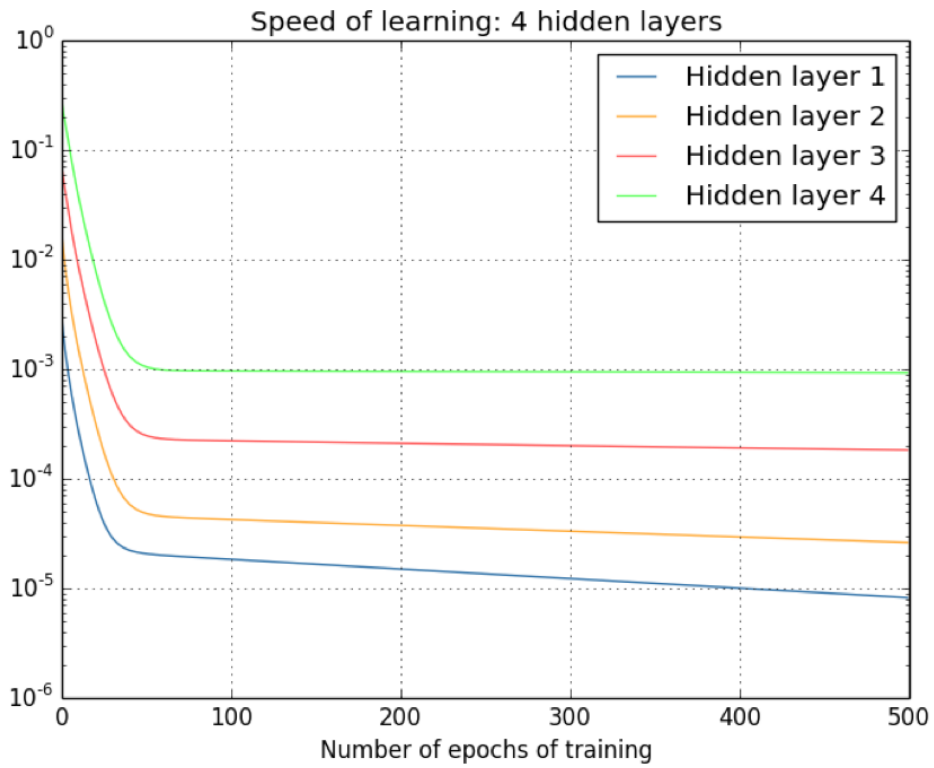
Luego de 500 épocas podemos comparar. Vemos que la curva amarilla toma el valor aproximado de 0,01. La azul toma el valor aproximado de 0,0025, o sea, un cuarto de la amarilla.

Ahora veamos que sucede si en lugar de dos capas ocultas utilizamos tres capas ocultas.



Ahora en la tercera capa el promedio del gradiente de los umbrales es aproximadamente **0,00102** (curva rosa), en la segunda capa el promedio es aproximadamente **0,00011** (curva amarilla) y en la tercera capa el,promedio es aproximadamente **0,00009** (curva celeste).

Veamos el mismo experimento pero para cuatro capas ocultas.



Ahora el promedio de los módulos de las componentes del gradiente de los umbrales en las diferentes capas varía entre aproximadamente 0,001 (curva verde) para la cuarta capa oculta y aproximadamente 0,000009 para la primera capa oculta (curva celeste).

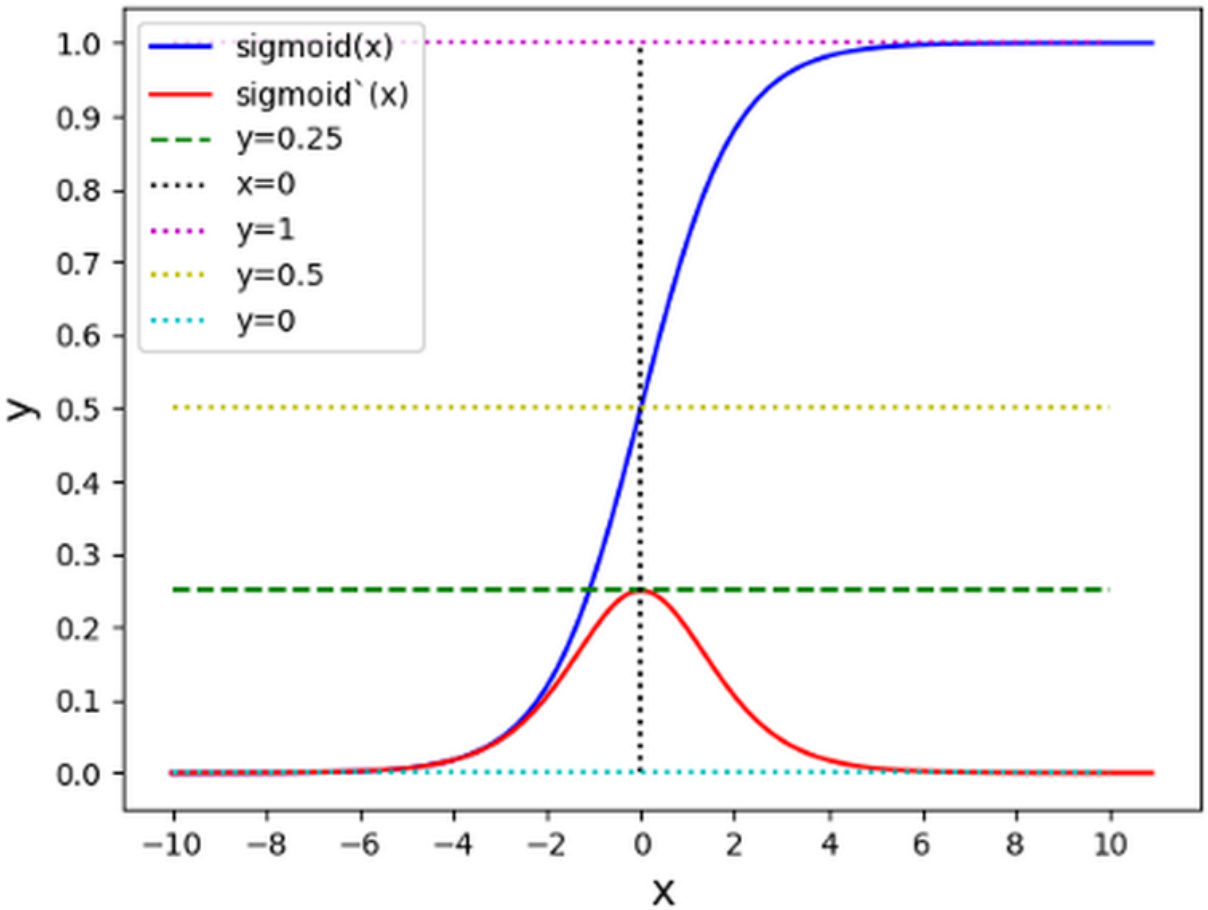
Observemos cómo disminuye el promedio de los módulos de la primera capa a medida que agregamos más capas.

Nro. capas oculta.	Promedio de la capa 1
2	0,0025
3	0,00009
4	0,000009

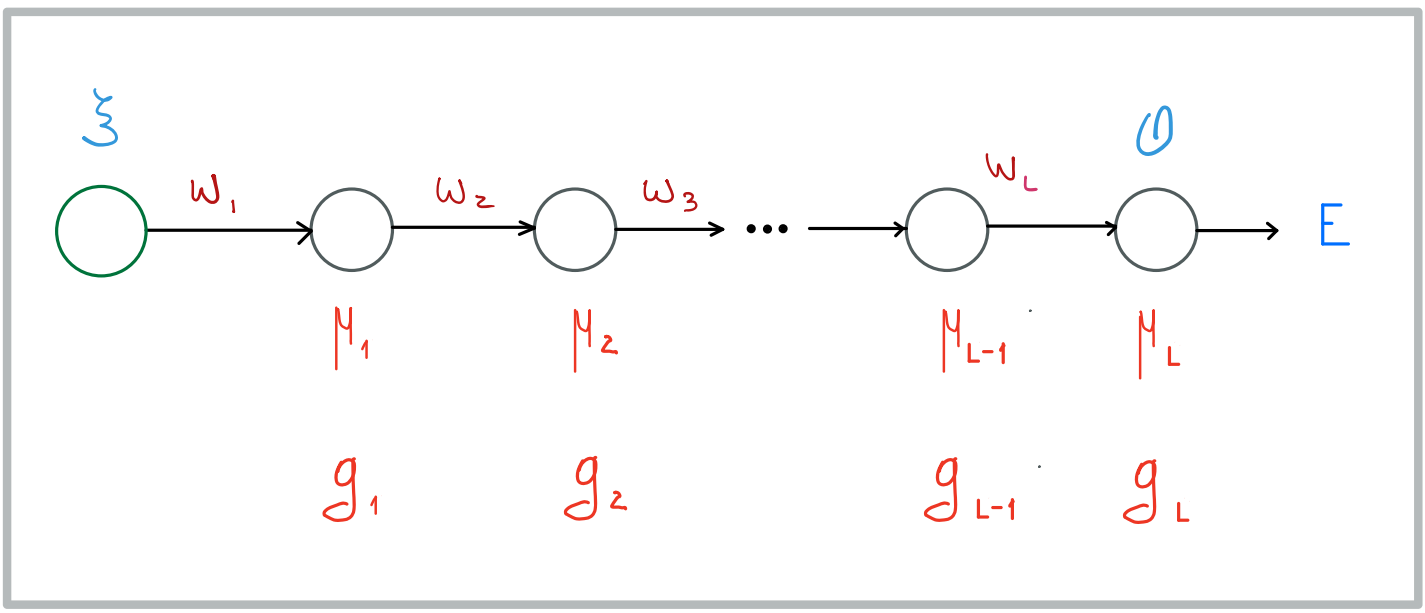
Si siguiésemos agregando capas, por cada capa disminuiría un orden de magnitud este promedio. Como usamos la misma razón de aprendizaje para todas las capas, estos promedios menos dan una idea de cómo se hace más y más lento el proceso de aprendizaje a medida que tenemos redes feed forward más y más profundas.

Esto nos dice que cuando ponemos muchas capas se hace imposible aprender. A este fenómeno se lo llama SUPRESIÓN DEL GRADIENTE.

Ya hemos identificado el origen de este pronblema: las derivadas exponencialmente pequeñas de las funciones de activación de las neuronas sigmoides o tangente hiperbólica.



Sigamos analizando un poco más el caso del descenso por el gradiente en su versión determinista. A fin de tener una percepción más didáctica del problema que queremos plantear, supongamos una red neuronal muy simple, con L capas y una neurona por capa. Veremos cómo aparece un fenómeno llamado SUPRESIÓN DEL GRADIENTE el cual afecta a las capas más cercanas a la entrada.



Es muy fácil generalizar este caso al caso de un número arbitrario de neuronas por capa. Repasemos el método del descenso por el gradiente. Presentamos todos los elementos del conjunto de entrenamiento y al final actualizamos las sinapsis de atrás hacia adelante. Omitiremos en esta etapa los umbrales.

$$w^L = w^L + \triangle w^L \qquad \triangle w^L = -\frac{\partial E}{\partial w^L}$$

$$w^{L-1} = w^{L-1} + \triangle w^{L-1} \qquad \triangle w^{L-1} = -\frac{\partial E}{\partial w^{L-1}}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$w^n = w^n + \triangle w^n \qquad \triangle w^n = -\frac{\partial E}{\partial w^n}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$w^1 = w^1 + \triangle w^1 \qquad \triangle w^1 = -\frac{\partial E}{\partial w^1}$$

$$\frac{\partial E}{\partial \omega_L} = g'_L(h_L) \times \left[\zeta^k - O^k\right]$$

$$\frac{\partial E}{\partial \omega_{L-1}} = g'_{L-1}(h_{L-1}) \times \omega_L \; g'_L(h_{L-1}) \times \left[\zeta^k - O^k\right]$$

$$\vdots \qquad\qquad \vdots$$

$$\frac{\partial E}{\partial \omega_K} = g'_K(h_K) \times \dots \times \omega_L g'_L(h_L) \times \left[\zeta^k - O^k\right]$$

$$\vdots \qquad\qquad \vdots$$

$$\frac{\partial E}{\partial \omega_2} = g'_2(h_2) \times \dots \times \omega_L g'_L(h_L) \times \left[\zeta^k - O^k\right]$$

$$\frac{\partial E}{\partial \omega_1} = g'_1(h_1) \times \omega_2 \times \dots \times \omega_L g'_L(h_L) \times \left[\zeta^k - O^k\right]$$

Vemos que cuanto más capas ocultas tenemos, las primeras capas tendrán más y más productos de derivadas, lo cual baja la magnitud promedio de los incrementos en cada capa.