QUANTIL SAS

Deep Learning: Teoría y Aplicaciones

16 de Octubre de 2014

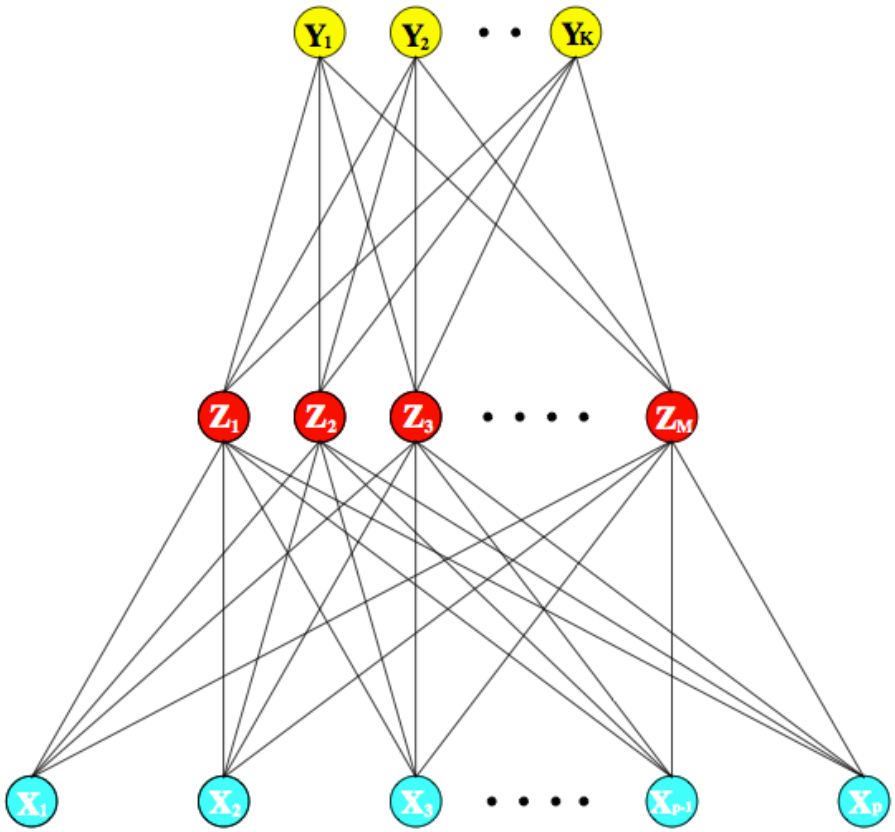
Motivación

- Se tiene un problema de clasificación con suficientes variables de predicción y suficientes datos.
- Las variables de predicción tienen un efecto conjunto no lineal sobre la clase. Por ejemplo, una foto.
- Se busca generar una representación más abstracta de las variables de predicción y con esta representación hacer una predicción.
- Ejemplos: Reconocimientos de imágenes y de voz, análisis técnico en finanzas.

Redes Neuronales

- Una red neuronal tiene varias capas. En la capa inicial, una neurona por cada variable con la que se quiere predecir la clase. Esta neurona recibe el dato correspondiente a esa variable y lo envía a la capa superior.
- En la capa final hay una neurona por cada clase, esta neurona toma las señales de la capa inferior y arroja la probabilidad con la que el dato pertenece a la clase.
- La capa inferior recibe los datos y los envía a la segunda capa, que convierte los datos en señales y los manda a la segunda capa, así sucesivamente.

Redes Neuronales



Redes Neuronales

- Suponga que nuestra red sólo tiene una capa intermedia. La capa inferior envía el vector X a cada neurona. Fíjese que todas las neuronas reciben la misma información. Esta neurona realiza una transformación lineal de la información recibida y la transforma en una señal usando una función logística

$$Z_m = \sigma(A_m X) \tag{1}$$

donde la matriz A_m tiene coeficientes distintos entre neuronas y que deberán ser ajustados más adelante. La función σ es una función logística dada por

$$\sigma(v) = 1/(1 + e^{-v}) \tag{2}$$

Redes Neuronales

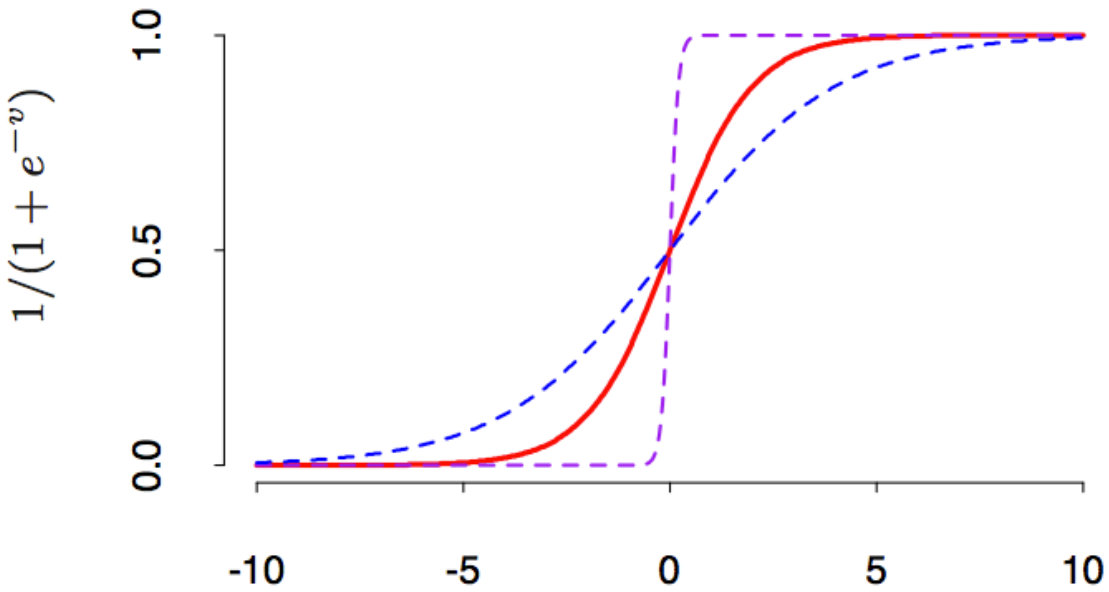
- Suponga que nuestra red sólo tiene una capa intermedia. La capa inferior envía el vector X a cada neurona. Fíjese que todas las neuronas reciben la misma información. Esta neurona realiza una transformación lineal de la información recibida y la transforma en una señal usando una función logística

$$Z_m = \sigma(A_m X) \tag{3}$$

donde la matriz A_m tiene coeficientes distintos entre neuronas y que deberán ser ajustados más adelante. La función σ es una función logística dada por

$$\sigma(v) = 1/(1 + e^{-v}) \tag{4}$$

Redes Neuronales



Redes Neuronales

- Para una red neuronal es muy fácil clasificar. Simplemente se ingresan los valores de las variables predictivas y se calculan las señales empezando por la capa inferior hasta llegar arriba.
- Entrenar una red en un proceso mucho más complejo. Para entrenar, es necesario determinar los coeficientes asociados a cada una de las neuronas.
- El proceso de entrenamiento es llamado Back-Propagation. Es simple: Se define una función de error y se minimiza sujeto a los parámetros de las neuronas. La minimización se realiza por descenso del gradiente.

$$R(\Theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} log(p_{ik}) \tag{5}$$

Redes Neuronales

Las redes neuronales presentan varios problemas

- La cantidad de variables libres que se deben entrenar es muy alta
- La función objectivo no es convexa en sus parámetros, por lo que el algoritmo de optimización de puede estancar en óptimos locales.
- La solución depende de la inicialización de los parámetros

Redes Neuronales

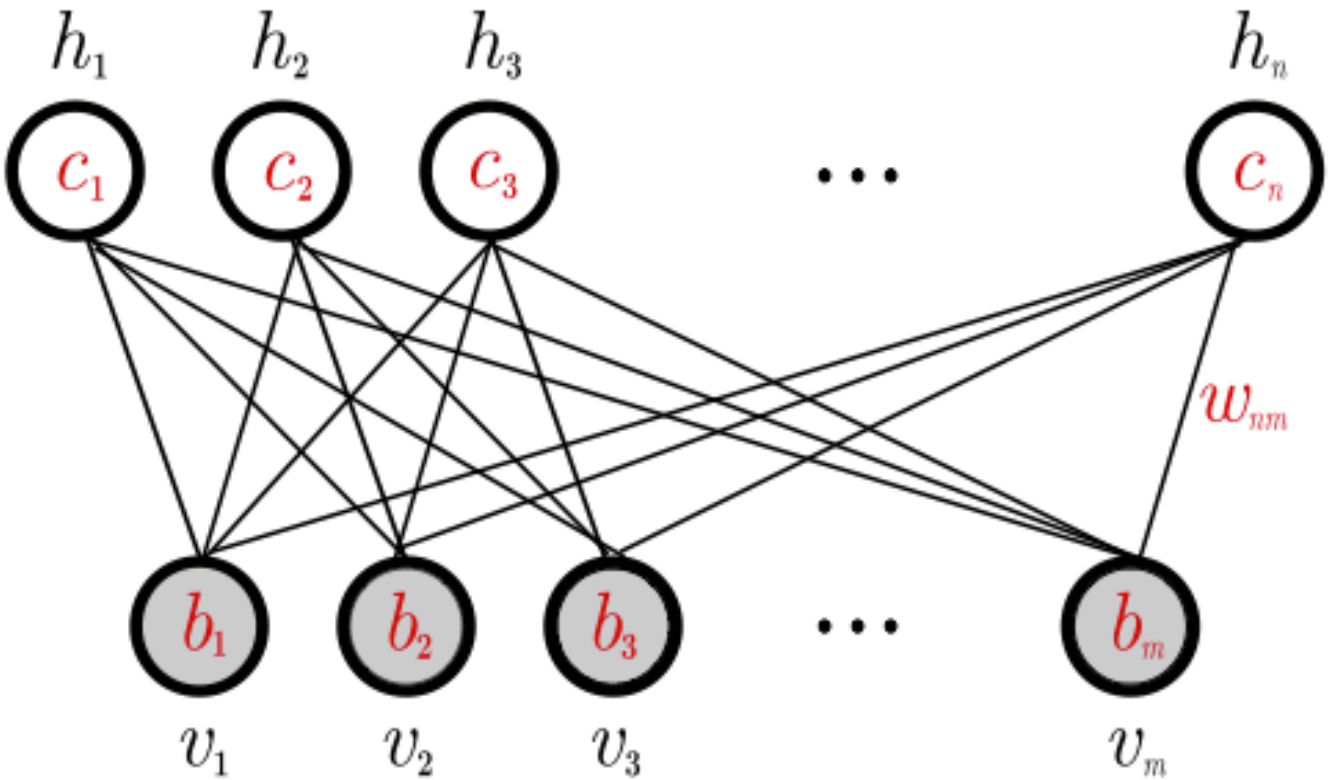
Los problemas tienen varias soluciones

- Convolutional Neural Networks, usadas para el reconocimiento de caracteres.
- Multiples inicializaciones de la red
- Máquinas de Boltzmann y Deep Belief Networks

Máquinas de Boltzmann

- Una máquina de Boltzmann es un grafo bipartito. Este grafo lo dividimos en dos grupos de nodos, nodos escondidos, que denotaremos con H y nodos visibles, que denotaremos por V.

Máquinas de Boltzmann



Máquinas de Boltzmann

- Una configuración de una máquina es un conjunto de valores que toman los nodos escondidos y los nodos visibles. Es decir, un vector de la forma $(v, h) \in \{0,1\}^V \times \{0,1\}^H$
- Además, se define una distribución de probabilidad sobre el conjunto de todos los estados posibles de la máquina. Esta distribución de probabilidad toma la forma

$$P(v, h) = \frac{e^{-Energy(v,h)}}{Z} \tag{6}$$

donde la función de energía es una función cuadrática dada por

$$Energy(v, h) = -b^T v - c^T h - h^T W v \tag{7}$$

donde es necesario estimar las constantes b, c y la matriz W.

Máquinas de Boltzmann

- Intuitivamente, una máquina de Boltzmann es una representación del mundo, los nodos visibles es la información que podemos ver, los nodos escondidos son una abstracción y son el proceso que genera los datos visibles.
- Ejemplo: Fotos de animales, tipos de acciones.

Ajuste de Máquinas de Boltzmann

- Una MB es esencialmente una distribución de probabilidad a la que hay que estimarle unos parámetros.
- Suponga que usted tiene unos datos y usted sabe que fueron generados por una distribución normal, pero usted necesita estimar la media y la varianza. Para eso usted usa sus datos y estima por máxima verosimilitud
- Aquí hay un problema, no tenemos datos de las variables escondidas, sólo de las variables visibles. Para resolverlo, calcule la distribución marginal.

Ajuste de Máquinas de Boltzmann

$$P(v) = \sum_{h} P(v, h) \tag{8}$$

- Esta es una distribución que únicamente depende de los datos de los nodos visibles y de los parámetros a estimar, así que se puede usar máxima verosimilitud.

Ajuste de Máquinas de Boltzmann

- El proceso por el que se maximiza la función de verosimilitud es descenso del gradiente, pero una versión muy particular. Al calcular el gradiente de la función de verosimilitud se obtiene lo siguiente

$$\frac{\partial log(P(v))}{\partial \theta} = -\sum_{h} P(v|h) \frac{\partial Energy(v,h)}{\partial \theta} + \sum_{v,h} P(v,h) \frac{\partial Energy(v,h)}{\partial \theta}$$

- El cálculo de este gradiente es computacionalmente imposible, así que se usa una versión de descenso del gradiente estocástica.
- El algoritmo, con el descenso estocástico y la metodología de sampleo, es conocido como Contrastive Divergence

Máquinas de Boltzmann

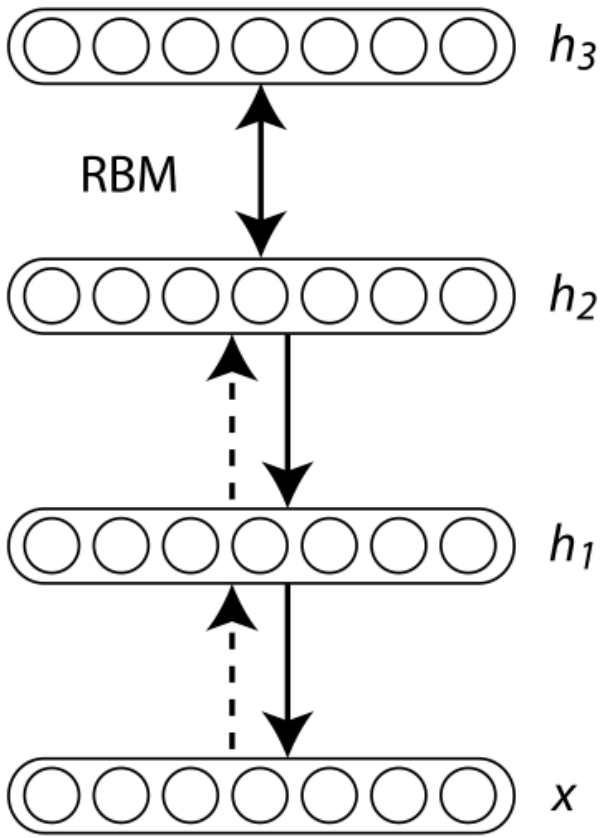
$$p(H_i = 1 \,|\, \boldsymbol{v}) = \sigma\left(\sum_{j=1}^{m} w_{ij} v_j + c_i \right)$$

$$p(V_j = 1 \,|\, \boldsymbol{h}) = \sigma\left(\sum_{i=1}^{n} w_{ij} h_i + b_j \right)$$

Deep Belief Networks

- Las máquinas de Boltzmann se pueden ordenar una encima de la otra para obtener una red similar a una red neuronal.
- Estas redes son conocidas como Deep Belief Networks y son usadas para modelar información visible usando distintas capas. Cada capa es un grado de abstracción de la información más alto.
- Por ejemplo, para construir una foto de un animal la primera capa puede abstraer el tipo de animal, la segunda puede abstraer la raza y el lugar, la siguiente capa el color, etc, y la última capa genera los pixeles de la imagen.

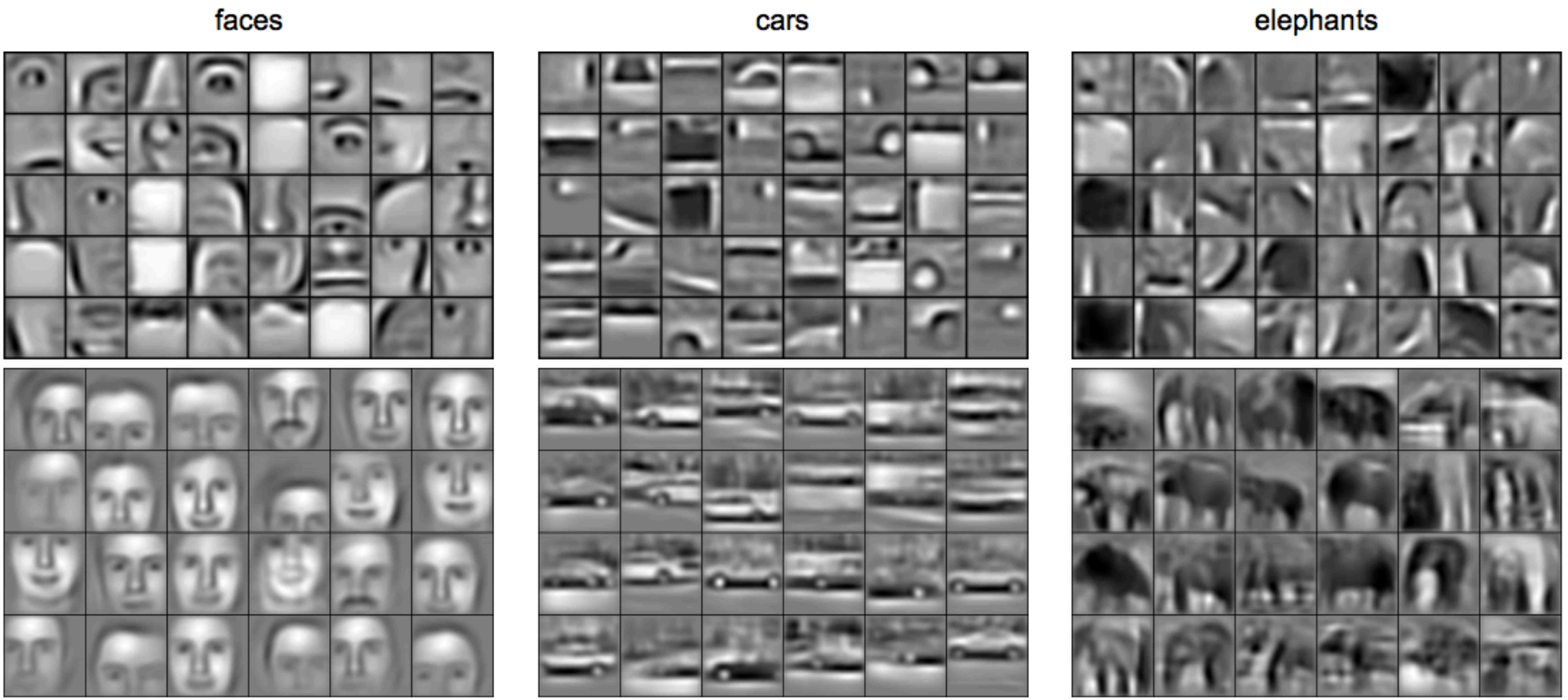
Deep Belief Networks



Aplicaciones

- Las Deep Belief Networks permiten hacer simulaciones de un fenómeno complejo. Por ejemplo, se puede entrenar una DBN con distintas caligrafías. El DBN entonces es capaz de abstraer características como la curvatura de las letras, el ancho, etc.
- Como un DBN es una distribución de probabilidad, se puede samplear de esta distribución y generar nuevas caligrafías. Estas nuevas caligrafías van a ser completamente originales y distintas a aquellas con las que se entrenó.
- Más importante, una DBN guarda mucha relación con una red neuronal. Es posible entrenar una DBN y después inicializar una red neuronal con los parámetros del DBN. Esta es la aplicación más importante y le dio una nueva vida al uso de redes neuronales, que habían sido hasta antes muy difíciles de entrenar

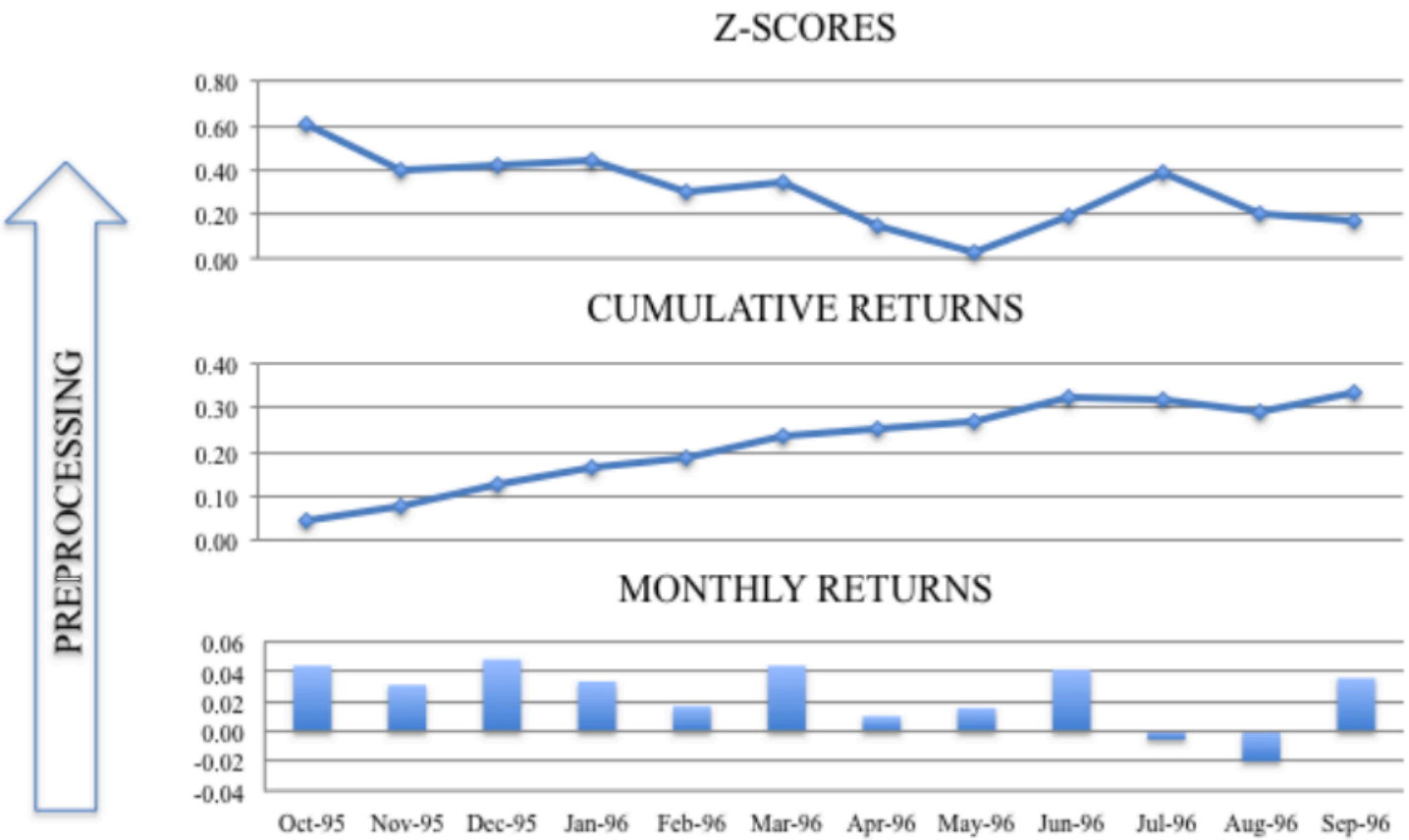
Aplicaciones



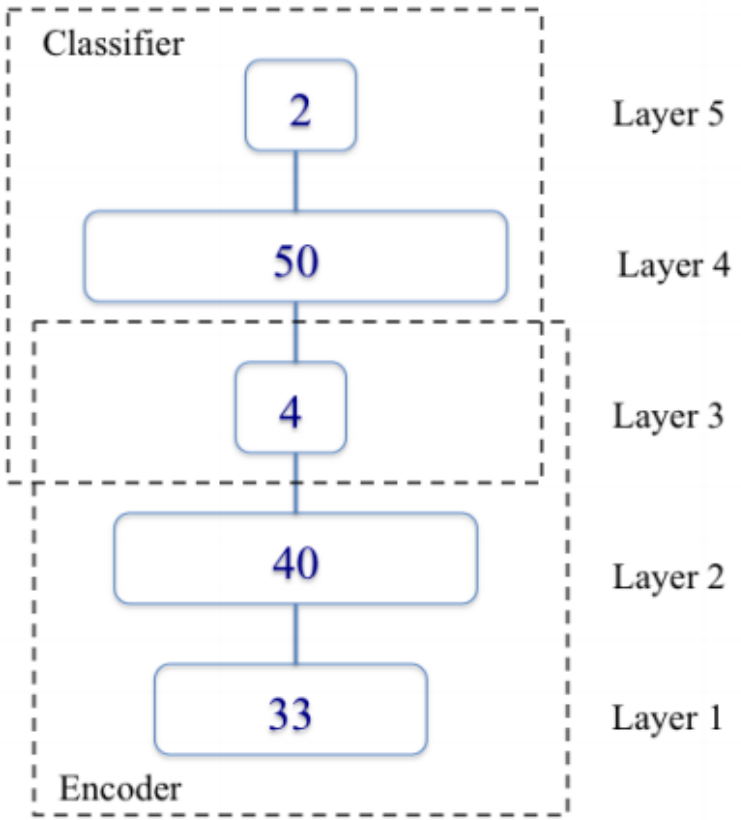
Enhance Momentum Trading Strategies

- Se observa que el precio de las acciones gana momento, esto se puede usar para generar estrategias.
- DL se usa normalmente para autimatizar tareas sencillas para los humanos, no es este el caso.
- En promedio hay 3.282 acciones cada mes. Se entrena con el período 1965-1989, se testea en 1990-2009
- Para cada mes se toman los retornos de los últimos 12 meses y de los últimos 20 días.
- Se calculan los retornos acumulados y se calcula un z-score a partir de un corte transversal de los datos.
- Se le da un label a los datos dependiendo de si el retorno está por encima de la mediana o por debajo de la mediana.

Preprocesamiento



Diseño de la Red



Diseño de la Red

- Se entrena una máquina de Boltzmann y una red neuronal simultáneamente.
- Se genera un cuello de botella para disminuir la dimensionalidad de la información.
- Se realiza hold-out cross validation para determinar el tamaño óptimo de la red.

Resultados Clasificación

Table 1. Confusion Matrix

		Predicted	
		1	2
Actual	1	22.38%	27.45%
	2	19.19%	30.97%

Estrategia de Trading

- Basados en las probabilidades de clasificación en la Clase 2 se dividen las acciones en aquellas ubicadas en el máximo decimal y en el mínimo decimal.
- Se diseña una estrategia en la que se compran acciones con alta probabilidad de Clase 2 y se va en corto en acciones con baja probabilidad de Clase 2.
- Se compara con una estrategia análoga en la que se toma la media de los retornos de los últimos 12 meses y allí se forman los deciles.

Resultados

Table 3. Stock characteristics by strategy.

	Decile 1	Decile 10	10 - 1
Past 12m ret			
- Enhanced	-0.39	0.23	0.62
- Basic	-1.05	2.03	3.08
Past 20d ret			
- Enhanced	0.41	-0.51	-0.92
- Basic	-0.06	0.05	0.11

Estrategia de Trading

- El algoritmo descubre que la mejor estrategia es comprar acciones que no siempre han venido subiendo, sino aquellas que iban subiendo pero últimamente no se han desempeñado muy bien.
- Esto es consistente con el short-term reversal effect
- Se obtienen retornos anuales del 43%