

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE INGENIERIA INFORMÁTICA

Propuesta de Tesis

Informe

Nombre: Felipe Alberto Reyes González

Programa: Magíster en Ingeniería Informática

Profesor patrocinante: Victor Parada

Cel.: 890 26 317

email: felipe.reyesg@usach.cl

21 de abril de 2017

Tabla de contenido

1. Introducción	3
1.1. Las redes neuronales	3
2. Algoritmo de retropropagación	4
3. El gradiente descendente	5
3.1. El desvanecimiento del gradiente	6

1 Introducción

El primer modelo de red neuronal artificiales (NN) fue propuesto en 1943 por McCulloch y Pitts en términos de un modelo computacional de actividad nerviosa. Las NN se han inspirado en las redes neuronales biológicas y las conexiones que construyen. Este modelo era un modelo binario, donde cada neurona posee un umbral, y sirvió de base para los modelos posteriores.

Las características principales de las NN son las siguientes:

1. Auto-organización y adaptabilidad: utilizan algoritmos de aprendizaje adaptativo y auto-organizativo, por lo que ofrecen mejores posibilidades de procesamiento robusto y adaptativo.
2. Procesado no lineal: aumenta la capacidad de la red para aproximar funciones, clasificar patrones y aumenta su inmunidad frente al ruido.
3. Procesado paralelo: normalmente se usa un gran número de nodos de procesamiento, con alto nivel de interconectividad.

1.1 Las redes neuronales

El elemento básico de las NN es el nodo, que recibe un vector de entrada para producir una salida como muestra en la figura 1. Cada entrada tiene asociado un vector de pesos w , que se va modificando durante el proceso de aprendizaje. Cada unidad aplica una función f sobre la suma de las entradas ponderada por el vector de pesos

$$y_i = \sum_j w_{ij} y_j$$

Donde el resultado puede servir como entrada de otras unidades.

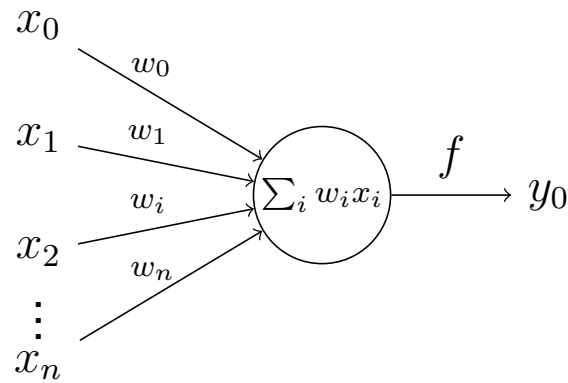


Figura 1: Neurona

Existen dos fases importante dentro del modelo

- Fase de entrenamiento: Se usa un conjunto de datos o patrones de entrenamiento para determinar los pesos que definen el modelo de la NN. Se calculan de manera iterativa, de acuerdo con los valores de entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la NN y la salida deseada.
- Fase de prueba: Durante el entrenamiento, el modelo se ajusta al conjunto de entrenamiento, perdiendo la habilidad de generalizar su aprendizaje a casos nuevos, a esta situación se le llama sobreajuste. Para evitar el sobreajuste, se utiliza un segundo grupo de datos diferentes, el conjunto de validación, que permitirá controlar el proceso de aprendizaje.

Los pesos óptimos se obtienen minimizando una función. Uno de los criterios utilizados es la minimización del error cuadrático medio entre el valor de salida y el valor real esperado.

2 Algoritmo de retropropagación

El problema de aprendizaje de una NN se formula en términos de la minimización de la función f de error asociada. Normalmente, esta función está compuesta por dos términos, uno que evalúa cómo se ajusta la salida de la red neuronal al conjunto de datos de que disponemos, que se denomina término *error*, y otro que se denomina *regulación*, y que se utiliza para evitar el sobreaprendizaje por medio del control de la complejidad efectiva

3 El gradiente descendente

El gradiente descendente busca los punto $p \in \Omega$ donde funciones del tipo $f : \Omega \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$ alcanzan su mínimo. La idea de este método se basa en que si f es una función diferenciable en todo su dominio Ω , entonces la derivada de f es un punto $p \in \Omega$ en dirección de un vector unitario $v \in \mathbb{R}^m$ se define como

$$df_p(v) = \nabla f(p)v$$

Observe que la magnitud de la ecuación es

$$|df_p(v)| = ||\nabla f(p)|| ||v|| \cos \theta = ||\nabla f(p)|| \cos \theta$$

Dicha magnitud es máxima cuando $\theta = 2n\pi, n \in \mathbb{Z}$. Es decir, para que $|df_p(v)|$ sea máxima, los vectores $\nabla f(p)$ y v debe ser paralelo. De esta manera, la función f crece más rápidamente en la dirección del vector $\nabla f(p)$ y decrece más rápidamente en la dirección del vector $-\nabla f(p)$. Dicha situación sugiere que la dirección negativa del gradiente $-\nabla f(p)$ es una buena dirección de búsqueda para encontrar el minimizador de la función f .

Sea $f : \Omega \subseteq \mathbb{R} \rightarrow \mathbb{R}$, si f tiene un mínimo en p , para encontrar a p se construye una sucesión de punto $\{p_t\}$ tal que p_t converge a p . Para resolver esto, comenzamos en p_t y nos desplazamos una cantidad $-\lambda_t \nabla f(p_t)$ para encontrar el punto p_{t+1} más cercano a p , es decir:

$$p_{t+1} = p_t - \lambda_t \nabla f(p_t)$$

donde λ_t se selecciona de tal manera que $p_{t+1} \in \Omega$ y $f(p_t) \geq f(p_{t+1})$

El parámetro λ_t se seleccionara para maximizar la cantidad a la que decrece la función f en cada paso.

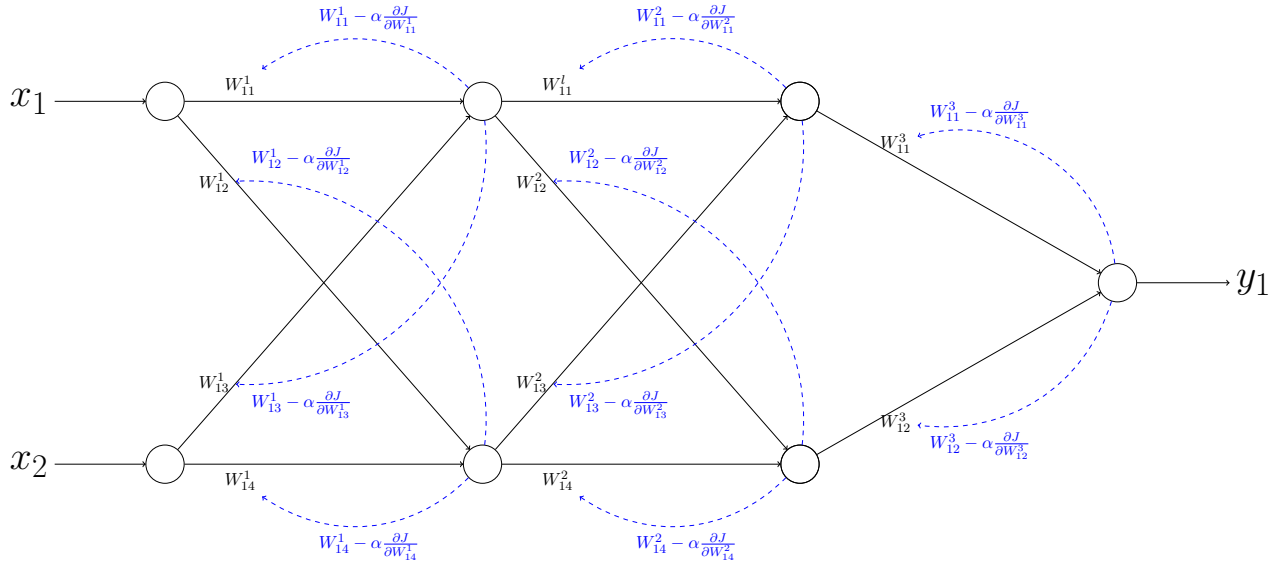


Figura 2: W^1_{ij} es el peso de la n -ésima neurona en la capa $l - 1$ a la j -ésima neurona de la capa l de la red.

3.1 El desvanecimiento del gradiente