

# CHAPTER 8: PRUNING NEURAL NETWORKS

## Resumen del capitulo

En capítulos anteriores se analizaron diferentes métodos de entrenamientos para redes neuronales, se realizó la implementación los distintos algoritmos que el autor presenta y obtuvieron las diferencias que caracterizan a cada uno de ellos. Para cada método de entrenamiento existente se tienen ventajas y desventajas, unos realizan cálculos muy precisos a costa de mucho procesamiento, mientras, que otros optimizan el tiempo de procesamiento ofreciendo un margen de error un tanto menor.

En este capítulo se abordaron dos nuevos algoritmos, poda selectiva y poda incremental, que tienen como función optimizar la estructura de la red neural. Ha diferencia de los algoritmos de entrenamiento los algoritmos de poda buscan modificar la estructura de la red neuronal impactando en menor medida la salida producida.

El objetivo principal de la poda es disminuir el procesamiento necesario para utilizar la red neuronal., aunque esto puede afectar la precisión de la salida de la red.

Se tienen dos tipos de poda:

* Poda Incremental

1. Se construye una red neuronal con la n cantidad de neuronas ocultas; n = 1 cuando el proceso se inicia por primera vez.
2. La red es entrenada con ayuda de un algoritmo de entrenamiento.
3. Por cada ciclo de entrenamiento se verifica el error,
   1. Si el error es aceptable la red tiene el número de neuronas ocultas adecuadas.
   2. De lo contrario pregunta si la cantidad de ciclos se completaron.
      1. La cantidad de neuronas ocultas aumenta en 1.
         1. Si la cantidad de neuronas ocultas llego al límite, significa que no se alcanzó el error deseado.
         2. Se pasa al paso 1.

* Poda selectiva

Para iniciar con la poda selectiva se debe contar con una red neuronal previamente entrenada.

* + 1. Se inicia n = 1.
    2. Se elimina la neurona n de la capa oculta y se evalúa la red.
       1. Si el error producido es aceptable, la neurona es eliminada de la red y se pasa al paso 1.
       2. De lo contrario n = n + 1
          1. Si n es menor al número de neuronas de la capa oculta se pasa al paso 2.
          2. De lo contrario, el método llego a su fin.

## Preguntas de revisión

1. **Describir los pasos del método de poda incremental.**
2. Se construye una red neuronal con la n cantidad de neuronas ocultas; n = 1 cuando el proceso se inicia por primera vez.
3. La red es entrenada con ayuda de un algoritmo de entrenamiento.
4. Por cada ciclo de entrenamiento se verifica el error,
   1. Si el error es aceptable la red tiene el número de neuronas ocultas adecuadas.
   2. De lo contrario pregunta si la cantidad de ciclos se completaron.
      1. La cantidad de neuronas ocultas aumenta en 1.
         1. Si la cantidad de neuronas ocultas llego al límite, significa que no se alcanzó el error deseado.
         2. Se pasa al paso 1.
5. **Describir los pasos del método de poda selectiva.**
6. Se inicia n = 1.
7. Se elimina la neurona n de la capa oculta y se evalúa la red.
   * Si el error producido es aceptable, la neurona es eliminada de la red y se pasa al paso 1.
   * De lo contrario n = n + 1
     1. Si n es menor al número de neuronas de la capa oculta se pasa al paso 2.
     2. De lo contrario, el método llego a su fin.
8. **Usted tiene una red neuronal que ya ha sido entrenada y produce buenos resultados. Sin embargo, te preguntas si podrías ser capaz de eliminar varias neuronas ocultas. ¿Qué método de poda usaría para lograr esto? ¿Por qué?**

Poda selectiva, la poda selectiva toma una red ya construida, va comprobando diversa configuración en la capa oculta, eliminando una a una las neuronas y verificando si el error producido es aceptable, en el caso que el error se aceptable la neurona se elimina de la red. Continua así hasta probar todas las posibles configuraciones de las neuronas de la red oculta.

1. **Le gustaría eliminar una neurona de la capa oculta. ¿Qué se debe hacer, en todo caso, con la matriz de peso entre la capa de entrada y la capa oculta? ¿Qué se debe hacer, en todo caso, con la matriz de peso entre la capa oculta y la capa de salida?**

Para la capa de entrada, que corresponde a la capa anterior a la capa oculta (hablando de que se podo un ejemplo de una red con una capa oculta), para eliminar los pesos de una neurona eliminada se debe eliminar la columna de pesos que corresponde a dicha neurona, de esta manera la red no se verá afectada.

Para la capa de salida es más sencillo, solo se debe eliminar la neurona. No se realiza ninguna acción extra.

1. **Qué método de poda debe utilizar uno de los algoritmos de aprendizaje como parte de su proceso.**

La poda incremental es utilizada junta a un método de entrenamiento. Esto se debe a que el algoritmo va probando con distinta cantidad de neuronas ocultas, iniciando con una, hasta que la red produzca el error aceptable, para cada incremento de neurona en la capa oculta, la red debe ser entrenada y verificar si puede alcanzar el error mínimo establecido.

## Vocabulario

**Connection Significance**

Conexión entre neuronas que no puede ser podada. Al podar algunas conexiones de la red, la salida no se ve tan afectada, pero eliminar una conexión significativa hace que se dispare el error y salga de los márgenes establecidos. Estas conexiones se pueden considerar como las conexiones fundamentales de la red.

**Incremental Pruning**

Algoritmo utilizado para optimizar la estructura de la red neuronal. Consiste en ir aumentando el número de neuronas de la capa oculta, iniciando con 1, y por cada aumento se lleva a cabo el entrenamiento de la red. Si red es capaz de alcanzar el error mínimo se conserva como la configuración más optima, de lo contrario el número de neuronas de la capa oculta aumenta en e1 y se repite el proceso de entrenamiento, esto se repite hasta llegar a la tase de error establecida o al llegar al número máximo de neuronas establecido.

**Pruning**

Proceso utilizado en redes neuronales para hacer que las redes neuronales sean más eficientes. Su principal objetivo es disminuir la cantidad de procesamiento necesario para una red neuronal.

**Selective Pruning**

Algoritmo utilizado para optimizar la estructura de la red neuronal. Consiste en ir eliminando neuronas de la capa oculta, se evalúa si el error que produce se encuentra dentro de los márgenes aceptados, si es así la configuración de la red pasa a ser la nueva red y el proceso se repite hasta que se hayan probado las diversas configuraciones en la capa oculta.

# CHAPTER 9: PREDICTIVE NEURAL NETWORKS

## Resumen del capitulo

Las redes neuronales feedfoward naturalmente se utilizar para el reconocimiento de patrones. Esta característica se puede utilizar para realizar predicciones. Para que la red neuronal sea capaz de realizar predicciones se debe proporcionar un conjunto de datos de entrenamiento. Después se le debe proveer los datos a predecir.

En el ejemplo que el autor propone se utiliza un *dataset* de 500 datos correspondientes a los valores de la función seno. Después el autor divide esos datos en datos de entrenamiento (250 datos) y en datos de prueba (250 datos). El método que el autor propone para el entrenamiento y las pruebas de la red es el siguiente:

Dado que el conjunto de datos proporcionado son valores consecutivos de la función seno, el autor opta por tomar los primero 5 valores consecutivos con las entradas y el 6 valor consecutivo como la salida, esto lo hace tanto como para el entrenamiento como para el testeo de la red.

El entrenamiento se realiza con el algoritmo *backpropagation,* de la misma forma que se ha comentado en capítulos anteriores.

Para el teste el autor toma el valor del ideal y la salida de la red y obtiene el RMS como indicador del error.

En las pruebas que se realizaron con la implementación con otra *dataset* (Mortalidad en Sinaloa), se observó una precisión en la mayoría de los casos, pero en una pequeña fracción de los datos el error se disparó.

En conclusión, la predicción realizada con la red neuronal feedforward puede ser factible cuando la variación de los valores de salida no es tan grande, como se alcanzó a observar en el ejemplo de la función seno, mientras que si se tiene una variación de los datos grande la red no es capaz de predecir con exactitud el valor, dado que los valores no tendrán ningún patrón aparente.

## Preguntas de revisión

1. **¿Se puede utilizar un mapa autoorganizado para la predicción? Si es así, ¿cómo configuraría las neuronas de entrada y salida?**

Si, la cantidad de neurona de entrada correspondería a la cantidad de datos a proporcionar, para la cantidad de neurona de salida se debe analizar la dimensión que puede tomar las posibilidades del problema, dado que si se establece un número inferior a las neuronas requeridas las predicciones carecerán de precisión, y si se elige un numero exagerado el tiempo de procesamiento de los datos será muy elevado.

Las redes neuronales SOM especialmente se usan para la clasificación de elementos, pero si se establece el numero adecuado de salidas (clases), podría ser utilizada para la predicción de datos, aunque tengo poco claro como seria el proceso para determinar el tamaño de la capa de salida.

1. **Cuando se utiliza una red neuronal feedforward para la predicción, ¿qué representan las neuronas de entrada? ¿Qué representan las neuronas de salida?**

Las neuronas de entrada representas espacios de tiempo conocidos, mientras que, las neuronas de salida representas espacios de tiempo desconocidos. El conjunto de estas se denomina bloque de tiempo y representa una porción de los datos. La correcta elección de la longitud de estos bloques hace que la red neuronal llegue a la tasa de error establecida.

1. **Tiene 500 intervalos de información. ¿Cómo podría utilizar estos datos para entrenar y probar una red neuronal predictiva?**

Una posible utilización de los datos seria tomar bloques de datos, donde el primer valore de dos bloques consecutivos, son también valores consecutivos.

Ejemplo

1 2 3 4 5 6 7 8 9

Entrenamiento

Entrada Salida

1 2 3 4

2 3 4 5

Testeo

5 6 7 8

6 7 8 9

De esta manera para los 500 datos.

Entrenamiento los primeros 250 valores, tomando bloques de 5, 4 entradas y 1 salida

Testeo los primeros 250 valores, tomando bloques de 5, 4 entradas y 1 salida

1. **¿Cómo elegiría entre el uso de una tangente hiperbólica y una función sigmoide como la función de activación para una red neuronal predictiva?**

Todo va en relación con el rango de valores que posee el dataset, si el dataset alcanza valores entre [1,-1] la función de activación tangente hiperbólica es la opción más factible, pero si los rangos de valores oscilan entre [0,1] la función de activación sigmoide es la adecuada.

Dado que no todos los valores estarán dentro de esos rangos, se debe considerar la normalización de los datos antes de ingresarlos a la red para que se produzcan valores adecuados. En la normalización de los datos se puede optar por dejar los rangos entre [0,1], o bien, [-1,1].

1. **¿Por qué podrías usar más de una neurona de salida en un trabajo predictivo de redes neuronales?**

El autor propone un ejemplo para la predicción con redes neuronales feedforwar, donde los datos son un conjunto consecutivo de valores, pero este no siempre será el caso. Se puede dar un *dataset* con distintos campos de información donde se tienen dos campos o más, que se desean predecir, en estos casos el uso de 2 o más neuronas de salida es necesario.

## Vocabulario

**Actual Data**

El autor trabaja con una clase llamada *ActualData* en ella se encuentras todos los métodos y variables necesarias para proveer los datos que se generaron por la función seno.

**Predictive Neural Network**

Red neuronal feedforwar que es utilizada para la predicción. Para que este tipo de red tenga una precisión estable en las predicciones que realiza, el conjunto de datos proporcionado no debe contener variación de datos grandes.

**Sine Wave**

La onda senoidal se produce a la hora de graficar los valores que se producen a partir de la función seno. Y tiene la característica entrar en un rango de valores [1,-1] para el eje y, mientras que para el eje x tiende a infinito.

**Temporal Neural Network**

Es otra forma de llamar a las redes neuronales predictivas.

Red neuronal feedforwar que es utilizada para la predicción. Para que este tipo de red tenga una precisión estable en las predicciones que realiza, el conjunto de datos proporcionado no debe contener variación de datos grandes.

**Time Slice**

Hacen referencia a los valores que se le proporciona a la red. El dataset se divide en bloques, denominados sectores de tiempo. Un sector de tiempo se compone de tiempos conocidos (neuronas de entrada) y de tiempo desconocidos (neuronas de salida). La correcta elección de la longitud de estos bloques garantiza una taza de error aceptable.