

Universidad TecMilenio

Nombre del alumno: Alexis Leal Mata

Máster en Inteligencia Artificial

Actividad 9

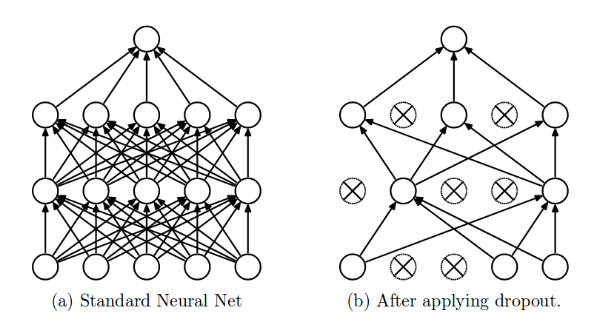
Técnicas de aprendizaje profundo

# Contesta el siguiente cuestionario

**El overfitting o sobreajuste es un fenómeno indeseado en un modelo de aprendizaje automático que provoca que sea bueno al trabajar con los datos de entrenamiento, pero no con datos nuevos o nunca vistos.**

**Para minimizar el error durante el entrenamiento se utilizan técnicas de regularización. Investiga sobre la estrategia de regularización DropOut utilizada en el aprendizaje profundo y explica con un ejemplo cómo funciona**

El término "DropOut" se refiere a la eliminación de los nodos (capa de entrada y oculta) en una red neuronal (como se ve en la Figura). Todas las conexiones hacia adelante y hacia atrás con un nodo descartado se eliminan temporalmente, creando así una nueva arquitectura de red a partir de la red principal. Los nodos se descartan con una probabilidad de abandono de p.



Por ejemplo, si las capas ocultas tienen 1000 neuronas (nodos) y se aplica un abandono con probabilidad de abandono = 0,5, entonces se eliminarán aleatoriamente 500 neuronas en cada iteración (lote).

**Define una estrategia para mejorar el desempeño de una red neuronal. Asegúrate de incluir alternativas de búsqueda de hiperparámetros, regularización y optimización.**

**Hiperarametros**: Los hiperparámetros son parámetros ajustables que permiten controlar el proceso de entrenamiento de un modelo. Por ejemplo, con redes neuronales, puede decidir el número de capas ocultas y el número de nodos de cada capa. El rendimiento de un modelo depende en gran medida de los hiperparámetros.

El ajuste de hiperparámetros, también denominado optimización de hiperparámetros es el proceso de encontrar la configuración de hiperparámetros que produzca el mejor rendimiento. Normalmente, el proceso es manual y costoso desde el punto de vista computacional

**Regularización**: La regularización en red consiste en penalizar de alguna forma las predicciones que hace nuestra red durante el entrenamiento, de forma que no piense que el training set es la verdad absoluta y así sepa generalizar mejor cuando ve otros datasets

Sus tipos son: L1/L2, Weight Decay, Dropout, Batch Normalization, Data Augmentation, Early Stopping

**Optimización**: optimizar los valores de los parámetros para reducir el error cometido por la red. El proceso mediante el cual se hace esto se conoce como “backpropagation”.

Los siguientes se encuentra disponibles en Python

Adagrad, Stochastic Gradient Descent (SGD), Adadelta, RMSprop, Adam, Nadam

**En un problema de aprendizaje profundo se te proporcionan 10,000,000 de ejemplos de datos. ¿Cuál sería una distribución adecuada de esos datos en los conjuntos de entrenamiento, validación y prueba?, ¿por qué?**

No existe una fórmula matemática que indique la forma correcta para la distribución de los datos para cada uno para hacer su respectivo test pero a mi opinión lo dejaría de la siguiente forma

Entrenamiento 70% (que sería la primera fase, en la parte de desarrollo de software lo vería cuando el programador hace pruebas unitarias)

validación 20% (seria la parte de QA)

prueba 10% (Seria por así decirlo la beta o pruebas con el usuario final)

Así tenemos mejores resultados y vemos donde están fallando

Si un modelo obtenido de una red neuronal tiene alto sesgo, ¿qué alternativas considerarías para corregirlo? Justifica tu respuesta.

Los ajustes de los hiperparametros nos ayudarían a solucionar una sesgo alto, pues esto cambiaria la forma que esta siendo entrenando la red

Unos ejemplos son los siguientes

Aumentar el número de capas ocultas.

Aumentar el número de unidades ocultas.

Entrenamiento para un mayor número de épocas.

Probando más redes neuronales.

Aumentar la cantidad de datos de entrenamiento.

Añadir normalización

Añadir de dropuot