

The background of the slide features a vector field represented by blue arrows and a contour plot with black lines. The arrows generally point from the left towards the right, with some local variations in direction. The contour lines form two main sets of nested loops, one in the upper right and one in the lower left, suggesting a complex underlying function or potential field.

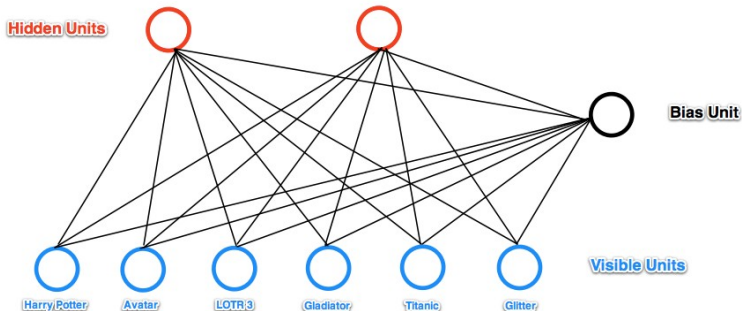
Deep Belief Networks

DBN

Redes Neuronales Profundas

Primer Cuatrimestre 2025

Restricted Boltzmann Machines



Una **Restricted Boltzmann Machine** (RBM) es una red neuronal artificial estocástica generativa que puede aprender la distribución de probabilidad de su dataset de entrada.

Restricted Boltzmann Machines: entrenamiento

Repetir hasta que el error entre el dataset de entrada y su reconstrucción caiga debajo de cierto umbral, o hasta alcanzar un número máximo de épocas:

1. Tomar una muestra v , computar las probabilidades de las neuronas ocultas, y muestrear un vector de activación h de esta distribución de probabilidad.
2. Computar el producto externo entre v y h y llamarlo gradiente positivo.
3. De h , reconstruir v^0 , y calcular h^0 de esto.
4. Computar el producto externo de v^0 y h^0 y llamarlo a esto el gradiente negativo.
5. Actualizar la matriz de pesos W con el gradiente positivo menos el gradiente negativo, multiplicado por el factor de aprendizaje: $\Delta W = (vh^T - v^0h^{0T})$
6. Actualizar los bias a y b según: $\Delta a = (v - v^0)$, $\Delta b = (h - h^0)$

Deep Belief Networks: definición

Una **Deep Belief Network** (DBN) es un modelo generativo probabilístico, compuesto de múltiples capas de variables latentes estocásticas, con conexiones entre las capas, pero no entre las unidades de cada capa. Las variables latentes tienen típicamente valores binarios y son llamados unidades ocultas o feature detectors. Los estados de las unidades en la primera capa representan el vector de entrada.

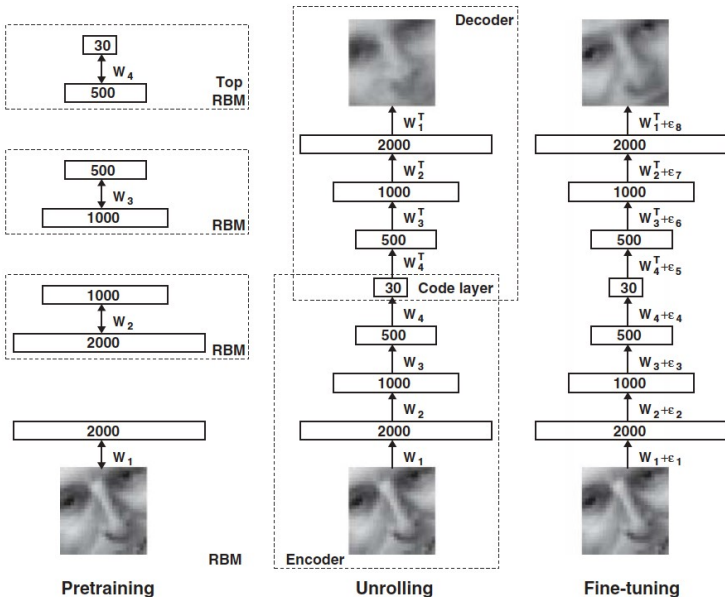
Las DBNs puede ser vistas como una composición de redes simples cada una de las cuales es una Restricted Boltzmann machine (RBMs), donde la capa oculta de cada subred, sirve como capa visible de la próxima red.

Deep Belief Networks: entrenamiento

Las DBNs son entrenadas una subred a la vez, empezando por la subred de abajo, tratando los valores de las variables latentes en una capa, como la capa visible para entrenar la próxima subred.

Luego de entrenar todas las subredes, se puede hacer un ajuste fino (fine-tuning) de los pesos, en forma supervisada mediante backpropagation.

Autoencoders y RBM: ejemplo

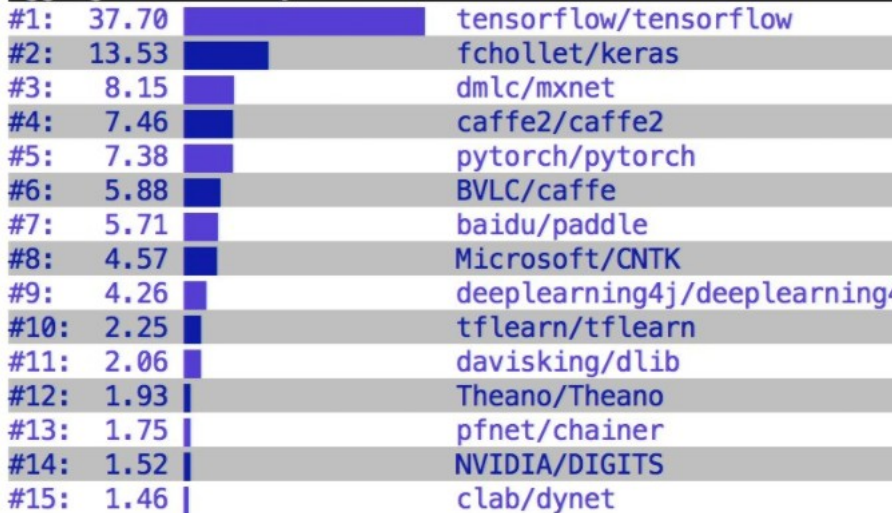


Deep Learning Frameworks: para qué?

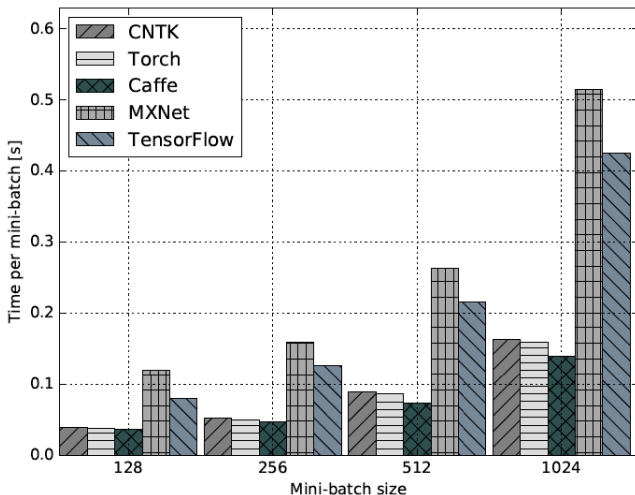
1. Uso transparente de la GPU (sólo debemos preocuparnos por programar el código de la red neuronal y el framework se encarga del manejo de la GPU).
2. Diferenciación automática.
3. Traen módulos de redes pre-implementadas, lo que nos ahorra tener que implementar todo de cero.

Deep Learning Frameworks: popularidad

aggregate activity from 2017-04-20 to 2017-07-06



Deep Learning Frameworks: velocidad (FCN)¹



¹ *Benchmarking State-of-the-Art Deep Learning Software Tools*, Shi, Wang, Xu, Chu, 2017. <https://arxiv.org/pdf/1608.07249.pdf>

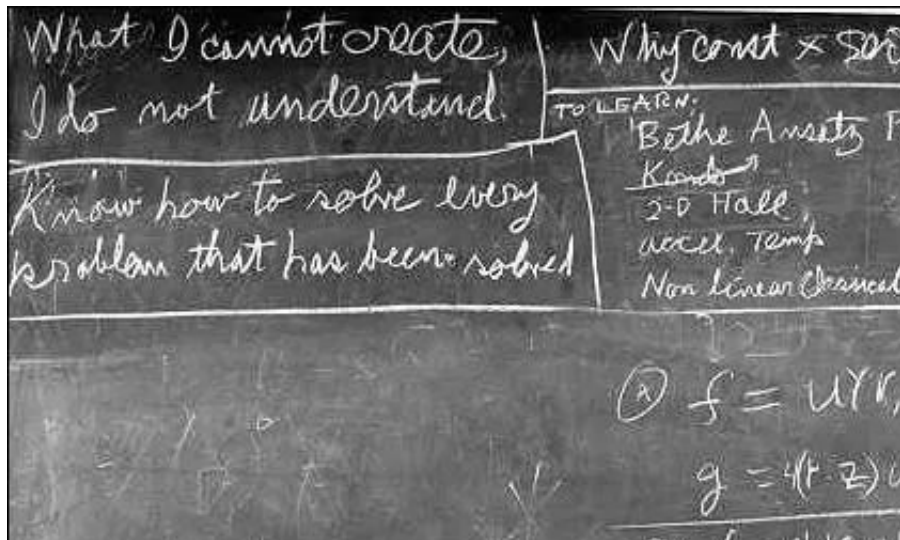
Pytorch vs. Tensorflow (hoy)

Pytorch es más rápido para hacer el prototipo, experimentar, debuguear. Hace el grafo computacional en tiempo de ejecución. Muy buena integración con Python!

Tensorflow puede ser más rápido para el producto final (en algunos casos). Hace el grafo computacional estático. Más difícil de debuguear.

Laboratorio: *What I cannot create, I do not understand.*

Richard P. Feynman



Ej 1 Pytorch: introducción

Realizar los siguientes tutoriales:

<http://pytorch.org/tutorials/>

Ej 2 Pytorch: ejemplos básicos

Entender el código del feedforward multicapa, y su implementación en Pytorch:

<https://github.com/jcjohnson/pytorch-examples>

Ej 3: The awesome pytorch list

Explorar tutoriales e implementaciones de papers en:

<https://github.com/bharathgs/Awesome-pytorch-list>