REDES NEURONALES 2024

Clase 22 parte 1

Jueves 7 de noviembre 2024

FAMAF, UNIVERSIDAD NACIONAL DE CÓRDOBA

INSTITUTO DE FÍSICA ENRIQUE GAVIOLA (UNC-CONICET)

2.- Las funciones exponenciales ralentizan el aprendizaje

Rectified Linear Units Improve Restricted Boltzmann Machines

Vinod Nair VNAIR@CS.TORONTO.EDU
Geoffrey E. Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Toronto, ON M5S 2G4, Canada

Abstract

Restricted Boltzmann machines were developed using binary stochastic hidden units. These can be generalized by replacing each binary unit by an infinite number of copies that all have the same weights but have progressively more negative biases. The learning and inference rules for these "Stepped Sigmoid Units" are unchanged. They can be approximated efficiently by noisy, rectified linear units. Compared with binary units, these units learn features that are better for object recognition on the NORB dataset and face verification on the Labeled Faces in the Wild dataset. Unlike binary units, rectified linear units preserve information about relative intensities as information travels through multiple layers of feature detectors.

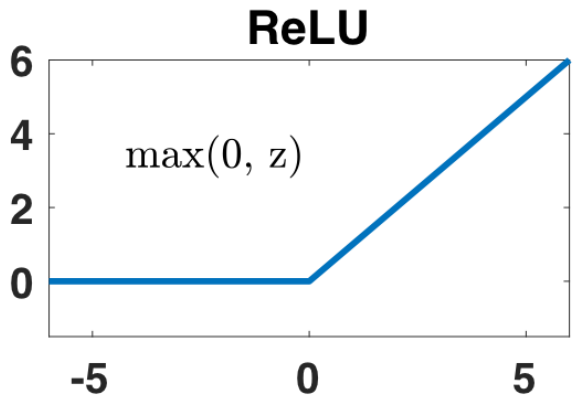
1.1. Learning a Restricted Boltzmann Machine

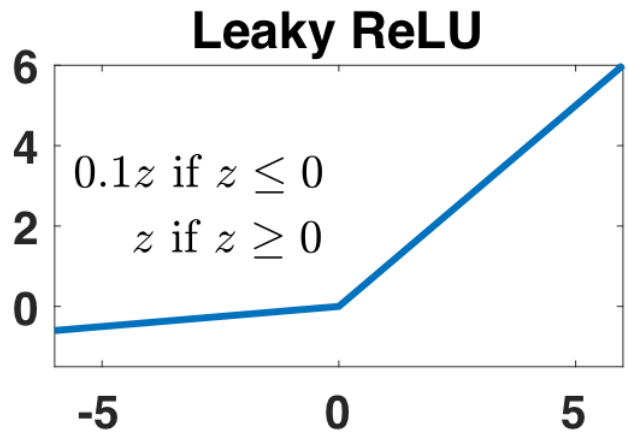
Images composed of binary pixels can be modeled by an RBM that uses a layer of binary hidden units (feature detectors) to model the higher-order correlations between pixels. If there are no direct interactions between the hidden units and no direct interactions between the visible units that represent the pixels, there is a simple and efficient method called "Contrastive Divergence" to learn a good set of feature detectors from a set of training images (Hinton, 2002). We start with small, random weights on the symmetric connections between each pixel i and each feature detector j. Then we repeatedly update each weight, w_{ij}, using the difference between two measured, pairwise correlations

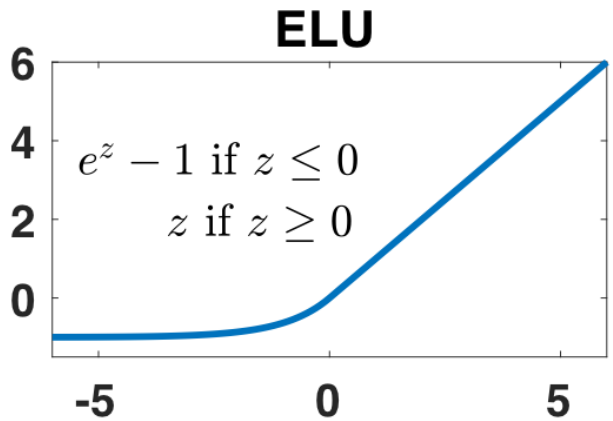
$$\Delta w_{ij} = \epsilon(<v_i h_j>_{data} - <v_i h_j>_{recon}) \qquad (1)$$

where ϵ is a learning rate, $<v_i h_j>_{data}$ is the frequency with which visible unit i and hidden unit j are on together when the feature detectors are being driven by

Cambiamos las funciones de activación para evitar la supresión del gradiente


ReLU
6
4
2
0
max(0, z)
-5
0
5


Leaky ReLU
6
4
2
0
$0.1z$ if $z \leq 0$
z if $z \geq 0$
-5
0
5


ELU
6
4
2
0
$e^z - 1$ if $z \leq 0$
z if $z \geq 0$
-5
0
5

3. Aprimoramos el descenso por el gradiente. DG estocástico

Descenso Gradiente

☑ **Minibatch:** Este algoritmo consiste en dividir el conjunto de entrenamiento en mini grupos por lo que en cada época se hace una reasignación aleatoria de todos los ejemplos de manera que $Qm = p$ (se tienen p elementos del conjunto de entrenamiento divididos en m grupos de tamaño Q). Con esta técnica se pretende introducir aleatoriedad en el método del descenso por el gradiente al sortear época a época los elementos del conjunto de entrenamiento. En general se encuentra que la cantidad de minibatches debe ser lo suficientemente alta para lograr introducir la aleatoriedad deseada, sin embargo si la cantidad de particiones es demasiado alta también lo será el costo computacional pues serán muchas veces que se realice el descenso por el gradiente. Con respecto a esto Keskar, Mudigere, Nocedal, Smelyanskiy, & Tang (2016) han comentado que cuando los batch son muy amplios (con muchos elementos de entrenamiento) la función costo adquiere una forma tal que el método del descenso por el gradiente tiende a converger hacia mínimos más agudos (pendientes altas). Por su parte, dividir los ejemplos en conjuntos de elementos más pequeños para analizar produce una convergencia del sistema hacia mínimos más planos (pendientes menores). Esto quiere decir que los batch más largos tienden hacia mínimos de los cuales es difícil despegarse mientras que los batch más pequeños, al ser atraídos hacia mínimos con formas más planas tienen la capacidad de escapar de las regiones de mínimos locales.

☑ **Dropout:** Para reducir el overfitting y permitir que una red neuronal artificial responda correctamente cuando es testeada, se aplica el dropout. Este algoritmo implica suprimir un cierto número de neuronas de manera aleatoria por lo que cuando se calcula la salida no se considera toda la información. Desde este punto de vista, cada neurona presente en las capas ocultas son omitidas al azar con una probabilidad $p = 0{,}5$ tal que estas neuronas no dependen del resto. Con esto se pretende evitar el establecimiento de complejas adaptaciones donde cada característica a detectar es útil y posible sólo en el contexto planteado por todas las demás. Es decir, cada neurona aprende a detectar una característica que permite producir la respuesta correcta en una multiplicidad de contextos internos posibles y variados en el cual opera la red. En sentido gráfico, el método del descenso por el gradiente no es capaz de salir de mínimos locales y requiere tiempos muy largos para salir de un punto de inflexión, tiempo que crece exponencialmente a medida que aumenta el número de componentes en la cual es plano. Este problema se facilita con la aplicación del algoritmo dropout al suprimir una de las dimensiones donde podría estar ubicado un mínimo local.

Es posible encontrar un paralelismo entre este funcionamiento de la red artificial (dropout) con el funcionamiento del cerebro dado que las neuronas requieren de un tiempo posterior a la emisión de señales para recuperar la energía liberada cuando las transmiten a neuronas vecinas de manera que, por un tiempo, se podría decir que permanecen apagadas.

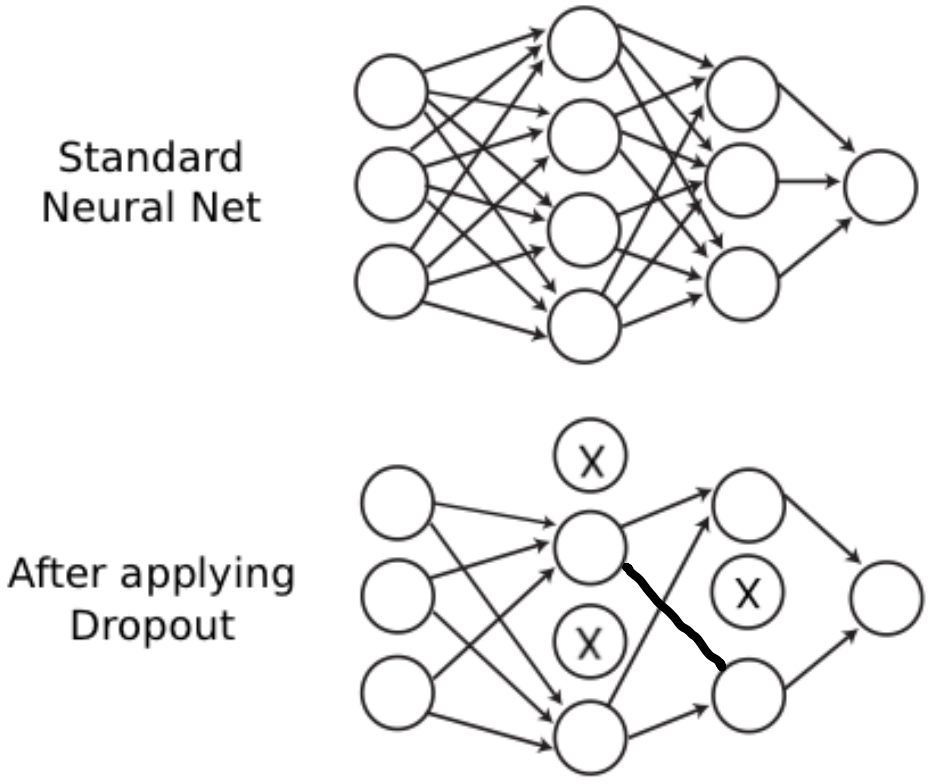


FIG. 39 **Dropout** During the training procedure neurons are randomly "dropped out" of the neural network with some probability p giving rise to a thinned network. This prevents overfitting by reducing correlations among neurons and reducing the variance in a method similar in spirit to ensemble methods.

*Improving neural networks by preventing co-adaptation of feature detectors by Geoffrey Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2012).