Deep Learning

by

AINDRILA GARAI

MSC STATISTICS, IIT KANPUR

aindrilag22@iitk.ac.in

Single Layer Neural Networks:

A neural network takes an input vector of p variables $X = (X_1, X_2, \ldots, X_p)$ and builds a nonlinear function $f(X)$ to predict the response Y. The neural network model has the form-

$$f(x) = \beta_0 + \sum_{k=1}^{K} \beta_k\, g\left(w_{k0} + \sum_{j=1}^{p} w_{kj}\, X_j\right)$$

where g(z) is a nonlinear activation function that is specified in advance. We can think of each $A_k = g\left(w_{k0} + \sum_{j=1}^{p} w_{kj}\, X_j\right)$ function as a different transformation $h_k(X)$ of the original features and $g(z) = \frac{e^z}{1+e^z}$

Note:

$$g(z) = (z)_+ = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases}$$

This ReLU (rectified linear ReLU unit) activation function stored more efficiently than a sigmoid activation.

- The final model is linear.
- The name neural network originally derived from thinking of these hidden units as analogous to neurons in the brain.
- The sum of two nonlinear transformations of linear functions can give us an interaction.

```
library (ISLR2)
```

Multilayer Neural Networks:

Modern neural networks typically have more than one hidden layer and often many units per layer and it may have more than one output. The first hidden layer is

$$A_k^{(1)} = g\left(w_{k0}^{(1)} + \sum_{j=1}^{p} w_{kj}^{(1)}\, X_j\right)$$

for $k = 1, \ldots, K_1$ and the second hidden layer is

$$A_\ell^{(2)} = g\left(w_{\ell 0}^{(2)} + \sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)}\right)$$

and $l = 1, \dots, K_2$ and so on.

Convolutional Neural Networks:

- Each image has a resolution of 32 × 32 pixels, with three eight-bit numbers per pixel representing red, green and blue. The numbers for each image are organized in a three-dimensional array called a feature map. The first two feature map axes are **spatial** (both are 32-dimensional), and the third is **channel axis** representing the three colors.

How does it work?

- CNNs mimic to some degree how humans classify images by recognizing specific features or patterns anywhere in the image that distinguish each particular object class. The network first identifies low-level features in the input image, such as small edges, patches of color, and the like.

- These low-level features are then combined to form higher-level features, the presence or absence of these higher-level features contributes to the probability of any given output class.

- A convolutional neural network combines two specialized types of hidden layers, called convolution layers and pooling layers.

- **Convolution layers:**

Convolution layers search for instances of small patterns in the image. A convolution filter relies on a very simple operation, called a convolution, which basically amounts to repeatedly multiplying matrix elements and then adding the results. The convolved image highlights regions of the original image that resemble the convolution filter. In a convolution layer, we use a whole bank of filters to pick out a variety of differently-oriented edges and shapes in the image.

- If we use K different convolution filters at this first hidden layer, we get K two-dimensional output feature maps, which together are treated as a single three-dimensional feature map

- **Pooling layers:**

Pooling layers downsample these to select a prominent subset. The max pooling operation summarizes each non-overlapping 2 × 2 block of pixels in an image using the maximum value in the block. This reduces the size of the image by a factor of two in each direction, and it also provides some location invariance.

- **Architecture of a Convolutional Neural Network:**

In a single convolution layer — each filter produces a new two-dimensional feature map. The number of convolution filters in a convolution layer is akin to the number of units at a particular hidden and defines the number of channels in the resulting threedimensional feature map. A pooling layer which reduces the first two dimensions of each three-dimensional feature map.

Noe:

- Each subsequent convolve layer is similar to the first. It takes as input the three-dimensional feature map from the previous layer and treats it like a single multi-channel image. Each convolution filter learned has as many channels as this feature map.
- Since the channel feature maps are reduced in size after each pool layer, we usually increase the number of filters in the next convolve layer to compensate.
- Sometimes we repeat several convolve layers before a pool layer. This effectively increases the dimension of the filter.
- These operations are repeated until the pooling has reduced each channel feature map down to just a few pixels in each dimension. At this point the three-dimensional feature maps are flattened — the pixels are treated as separate units — and fed into one or more fully-connected layers before reaching the output layer, which is a softmax activation for the 100 classes.

Document Classification:

Each document can be a different length, include slang or non-words, have spelling errors, etc. We need to find a way to featurize such a document. The simplest and most common featurization is the bag-of-words model. We score each document for the presence or absence of each of the words in a language dictionary. If the dictionary contains M words, that means for each document we create a binary feature vector of length M, and score a 1 for every word present, and 0 otherwise. That can be a very wide feature vector.

- A lasso logistic regression using the glmnet package- The lasso sequence is indexed by the regularization parameter λ
- A two-class neural network with two hidden layers, each with 16 ReLU units- The neural-net sequence is indexed by the number of gradient-descent iterations used in the fitting, as measured by training epochs or passes through the training set.

A two-class neural network amounts to a nonlinear logistic regression model is-

$$\log\left(\frac{\Pr(Y=1\mid X)}{\Pr(Y=0\mid X)}\right) = Z_1 - Z_0 = (\beta_{10} - \beta_{00}) + \sum_{\ell=1}^{K_2} (\beta_{1\ell} - \beta_{0\ell})\, A_\ell^{(2)}$$

There are two ways to summarize the words ignoring the context is- 1. A bag of 2-grams records the consecutive co-occurrence of every distinct pair of words. “Blissfully long” can be seen as a positive phrase in a movie review, while “blissfully short” a negative. 2. Treat the document as a sequence, taking account of all the words in the context of those that preceded and those that follow

Recurrent Neural Networks:

- In a recurrent neural network (RNN), the input object X is a sequence of words. The order of the words and closeness of certain words in a sentence convey semantic meaning. RNNs are designed to accommodate and take advantage of the sequential nature of such input objects, much like convolutional neural networks accommodate the spatial structure of image inputs. The output Y can also be a sequence.
- One can also have additional hidden layers in an RNN. For example, with two hidden layers, the sequence Aℓ is treated as an input sequence to the next hidden layer in an obvious fashion.
- The RNN we used scanned the document from beginning to end; alternative bidirectios.

When to Use Deep Learning:

image classification such as machine diagnosis of mammograms or digital X-ray images, ophthalmology eye scans, annotations of MRI scans, and also in speech and language translation, forecasting, and document modeling.

Fitting a Neural Network:

The parameters are $\beta = (\beta_0, \beta_1, \ldots, \beta_K)$ as well as each of the $w_k = \left(w_{k0}, w_{k1}, \ldots, w_{kp}\right)$, $k = 1, \ldots, K$. Given observations (x_i, y_i), $i = 1, \ldots, n$, we could fit the model by solving a nonlinear least squares problem.

$$\underset{\{w_k\}_1^K, \beta}{\text{minimize}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2,$$

where

$$f(x_i) = \beta_0 + \sum_{k=1}^{K} \beta_k \, g\left(w_{k0} + \sum_{j=1}^{p} w_{kj} \, x_{ij}\right).$$

- By Backpropagation the act of differentiation assigns a fraction of the residual to each of the parameters via the chain rule.

- When n is large, we can sample a small fraction or minibatch of them each time we compute a gradient step. This process is known as stochastic gradient descent (SGD).

Network Tuning:

- The number of hidden layers, and the number of units per layer. Modern thinking is that the number of units per hidden layer can be large, and overfitting can be controlled via the various forms of regularization.

- Regularization tuning parameters. These include the dropout rate ϕ and the strength λ of lasso and ridge regularization, and are typically set separately at each layer.