Neural Networks in Machine Learning

Matthijs Van keirsbilck

October 9, 2017

1 Overview

Neural networks are inspired on the brain, and are a technique to automatically learn complex input-output mappings in order to classify certain data. They can achieve superhuman performance in certain tasks.

1. Fully Connected Networks: Just a bunch of neurons, possibly in several layers. Neurons in a layer are connected to all the neurons in the next layer. These can complex imput-output mappings and are used for classification (eg you give an input, it tells you what the input is. For images, eg a dog or a car). Because there are so many connections, this does not scale at all. Therefore the input data is often preprocessed in certain ways to reduce dimensionality. This can be done in the network itself as well, for example by a CNN.

2. Neural networks are trained with gradient descent through backpropagation, a technique used for optimizing the connections between the neurons (the 'weights') in such a way that the outputs of the network best correspond with the training data (so you try to reduce the error). Weights that contribute a lot to the error (meaning the weight value is bad) will be modified a lot, while weights that don't contribute to the error (meaning they're good weights), are not changed.

3. Visual image processing deals with high-dimensional input data. To reduce the dimensionality, many people use Convolutional Neural Networks (CNNs). Example networks: AlexNet, VGG (fairly simple), and more complex networks like ResNet, Inception etc.

4. Time information (sequences). Eg for speech the temporal information (what follows what) is important to improve your predictions. To use this temporal information, Recurrent Neural Networks (RNNs) are used. A more complex variant are GRU and LSTM networks. See below for tutorials/explanation. This can also apply to visual processing in videos (one frame contains lots of info about what will be in the next frame); then CNNs and RNNs can be combined.

2 Recommended Reading

- Quick intro
- Stanford course on Convolutional Networks for visual processing. Very recommended for general Neural Network techniques and background (how to setup a network, how training works, what parameters and options you have during training etc). Also videos on YouTube available
- Very good blog page by a Machine Learning researcher, very informative and educational
- TensorFlow tutorial for MNIST (handwritten number recognition). If something is not clear, you can find a lot more information in the next few links
- Recurrent neural networks example of what RNNs can do for text processing
- TensorFlow implementation
- TensorFlow tutorial for LSTMs on MNIST
- If you really want to understand how a neural network works and how to implement it, this tutorial implements it from scratch. Very well written.

3 Neural Network Frameworks

You don't need to manually code everything in Matlab, other people have taken care of much of the general setup and created some really good frameworks (see below).

OS: Most things work on Windows, but sometimes it's a bit more complex to set up than on Linux. You might come across examples that don't support Windows, most researchers use Linux.

IDE: For Python I think PyCharm is super good. It has a built-in terminal which is really cool.

Python: Python is pretty much the default machine learning language. Use Anaconda to install packages in some sort of 'virtual machine' so that your system doesn't get messed up. I recommend to use TensorFlow. It's powerful without being overwhelming or hiding the underlying workings, it's backed by Google and nowadays has the largest community, so lots of examples and tutorials are available. If you have a NVIDIA gpu with CUDA, make sure to install tensorflow-gpu, not just tensorflow! Using the GPU can make running your networks 10+ times faster.

1. Theano: one of the first frameworks. Very low-level, meaning it gives you a lot of power. Can be difficult to use at first. There are wrappers that make this easier. Lasagne is a good one, though very light so there

are some things you'll need to implement yourself. Another one is Keras, which is very high-level so easy to get working but you don't get to really see what's going on under the hood. Keras can also use TensorFlow as backend, which is really cool. Theano will be discontinued soon.

2. Torch: also a pretty early one. I don't have much experience with it.

3. TensorFlow: currently probably the most popular framework. Quite fast nowadays, easy to use but still allowing for lots of custom low-level modifications. Backed by Google, so future-proof.

4. Caffe: also a big framework in research, especially in the hardware- neural network area (eg. Deep Compression)

3.1 Datasets

1. Images

   - MNIST: handwitten digit recognition. Small dataset, easy to train etc. Standard for simple examples and tutorials
   - CIFAR10: simple image classification. Fairly small dataset, easy to train etc. Also lots of tutorials available.
   - ImageNet: large dataset (100+ GB), standard benchmark for serious image classification in research.

2. Speech

   - TIMIT: speech recognition dataset with sentence-level, word-level and phoneme-level labels. Non-free. My repo with code for TIMIT.
   - TCD-TIMIT: audio-visual speech dataset (400+ GB) with labeling similar to TIMIT. Requires lots of preprocessing (only raw video files and label files are provided. You could use this repository for that.
   - Voxforge
   - Lip Reading Sentences (LRS)
   - GRID lipreading dataset

4 Interesting Papers

4.1 Visual

- Krizhevsky et al. 2012, AlexNet
- He et al, 2014 Residual Neural Networks

4.2 Speech

- Deepmind: WaveNet
- Baidu, Deep Speech
- Oxford, Deepmind: lipreading with LipNet
- Uni Oxford/ Deepmind: audio-visual speech recognition (LipNet follow-up)

4.3 Reinforcement Learning

- Deepmind, AlphaGo
- A. Karpathy, learning Pong from Pixels

5 Other stuff

- Stanford Course for language processing (background info). The Visual course is better I think
- Page with lots of interesting papers
- Wikipedia is always good (donate :))
- Another good overview of CNNs, with links to important papers in the field
- Overview of (publicly) available datasets
- For training with GPUs, the GTX 1060 (6GB) or 1070 are good in terms of price-performance