Chapter – 0

**Neural Networks for Machine Learning**

**Geoffrey Hinton**

with **Nitish Srivastava** & **Kevin Swersky**

**Course Overview**

|  |  |
| --- | --- |
| **Chapter 1 : What are Neural Networks?**  1.1 — Why do we need Machine Learning  1.2 — What are Neural Networks  1.3 — Some simple Models Of Neurons  1.4 — A simple Example of learning  1.5 — Three Types of learning  **Chapter 2: Perceptrons**  2.1 — Types of neural network architectures  2.2 — Perceptrons first generation neural networks  2.3 — A geometrical view of perceptrons  2.4 — Why the learning works  2.5 — What perceptrons cant do  **Chapter 3 : Weights & Backpropagation**  3.1 — Learning the weights of a Linear Neuron  3.2 — The error surface for a Linear Neuron  3.3 — Learning weights of Logistic Output Neuron  3.4 — The BACKPROPAGATION algorithm  3.5 — Using the derivatives from backpropagation  **Chapter 4 : Outputs**  4.1 — Learning to predict the next word  4.2 — A brief diversion into Cognitive Science  4.3 — The SOFTMAX output function  4.4 — Neuro Probabilistic Language Models  4.5 — Dealing with many possible outputs  **Chapter 5 : Convolutional Nets**  5.1 — Why object recognition is difficult  5.2 — Achieving Viewpoint Invariance  5.3 — Convolutional Nets for digit recognition  5.4 — Convolutional Nets for object recognition  **Chapter 6 : Gradient Descent**  6.1 — Overview of Mini Batch Gradient Descent  6.2 — A bag of tricks for mini batch gradient descent  6.3 — The Momentum Method Neural  6.4 — Adaptive Learning Rates for each connection  6.5 — Rmsprop normalize the gradient  **Chapter 7 : RNNs**  7.1 — Modeling Sequences a brief overview  7.2 — Training RNNs with Back Propagation  7.3 — A toy example of Training an RNN  7.4 — Why it is Difficult to Train an RNN  7.5 — Long Term Short Term Memory  **Chapter 8 : Hessian free optimization**  8.1 — A brief overview of Hessian free optimization  8.2 — Modeling character strings  8.3 — Predicting the next character using HF  8.4 — Echo State Networks | **Chapter 9 : Regularization**  9.1 — Overview of ways to *improve generalization*  9.2 — *Limiting the size* of the *weights*  9.3 — Using noise as a regularizer  9.4 — Introduction to the Full Bayesian Approach  9.5 — The Bayesian interpretation of weight decay  9.6 — *MacKay's quick and dirty method*  **Chapter 10 : Bayesian Learning**  10.1 — Why it helps to combine models  10.2 — Mixtures of Experts  10.3 — The idea of Full Bayesian Learning  10.4 — Making full Bayesian learning practical  10.5 — Dropout  **Chapter 11 : Hopfield Nets**  11.1 — Hopfield Nets  11.2 — Dealing with spurious minima  11.3 — Hopfield nets with hidden units  11.4 — Using stochastic units to improve search  11.5 — How a Boltzmann machine models data  **Chapter 12 : RBM**  12.1 — Boltzmann machine learning  12.2 — More efficient ways to get the statistics  12.3 — Restricted Boltzmann Machines  12.4 — An example of RBM learning  12.5 — RBMs for collaborative filtering  **Chapter 13 : Belief Nets (DBN)**  13.1 — The *ups and downs* of BACKPROPAGATION  13.2 — Belief Nets  13.3 — Learning sigmoid belief nets  13.4 — The wake sleep algorithm  **Chapter 14 : Stacked RBM**  14.1 — Learning layers of features by stacking RBMs  14.2 — Discriminative learning for DBNs  14.3 — Discriminative fine tuning  14.4 — Modeling real valued data with an RBM  14.5 — RBMs are infinite sigmoid belief nets  **Chapter 15 : Autoencoders**  15.1 — From PCA to autoencoders  15.2 — Deep autoencoders  15.3 — Deep autoencoders for document retrieval  15.4 — Semantic Hashing  15.5 — Learning binary codes for image retrieval  15.6 — Shallow autoencoders for pre training  **Chapter 16 : Optimization**  16.1 — Learning a joint model of images and captions  16.2 — Hierarchical Coordinate Frames  16.3 — Bayesian optimization of hyper parameters  16.4 — The fog of progress |