

CS231n Convolutional Neural Networks for Visual Recognition

These notes accompany the Stanford CS class [CS231n: Convolutional Neural Networks for Visual Recognition](#).

For questions/concerns/bug reports contact [Justin Johnson](#) regarding the assignments, or contact [Andrej Karpathy](#) regarding the course notes. You can also submit a pull request directly to our [git repo](#).

We encourage the use of the [hypothes.is](#) extension to annotate comments and discuss these notes inline.

Spring 2019 Assignments

Assignment #1: Image Classification, kNN, SVM, Softmax, Neural Network

Assignment #2: Fully-Connected Nets, Batch Normalization, Dropout, Convolutional Nets

Assignment #3: Image Captioning with Vanilla RNNs, Image Captioning with LSTMs, Network Visualization, Style Transfer, Generative Adversarial Networks

Module 0: Preparation

Setup Instructions

Python / Numpy Tutorial

IPython Notebook Tutorial

Google Cloud Tutorial

AWS Tutorial

Module 1: Neural Networks

Image Classification: Data-driven Approach, k-Nearest Neighbor, train/val/test splits
[L1/L2 distances](#), [hyperparameter search](#), [cross-validation](#)

Linear classification: Support Vector Machine, Softmax
[parametric approach](#), [bias trick](#), [hinge loss](#), [cross-entropy loss](#), [L2 regularization](#), [web demo](#)

Optimization: Stochastic Gradient Descent
[optimization landscapes](#), [local search](#), [learning rate](#), [analytic/numerical gradient](#)

Backpropagation, Intuitions
[chain rule interpretation](#), [real-valued circuits](#), [patterns in gradient flow](#)

Neural Networks Part 1: Setting up the Architecture
[model of a biological neuron](#), [activation functions](#), [neural net architecture](#), [representational power](#)

Neural Networks Part 2: Setting up the Data and the Loss

preprocessing, weight initialization, batch normalization, regularization (L2/dropout), loss functions

Neural Networks Part 3: Learning and Evaluation

gradient checks, sanity checks, babysitting the learning process, momentum (+nesterov), second-order methods, Adagrad/RMSprop, hyperparameter optimization, model ensembles

Putting it together: Minimal Neural Network Case Study

[minimal 2D toy data example](#)

Module 2: Convolutional Neural Networks

Convolutional Neural Networks: Architectures, Convolution / Pooling Layers

[layers](#), [spatial arrangement](#), [layer patterns](#), [layer sizing patterns](#), [AlexNet/ZFNet/VGGNet case studies](#), [computational considerations](#)

Understanding and Visualizing Convolutional Neural Networks

[tSNE embeddings](#), [deconvnets](#), [data gradients](#), [fooling ConvNets](#), [human comparisons](#)

Transfer Learning and Fine-tuning Convolutional Neural Networks

 [cs231n](#)

 [cs231n](#)

karpthy@cs.stanford.edu