2、What Is Deep Learning

specialize in 专门研究~

be around for 存在

hierarchical learning 层学习：where simple concepts are learned in the lower layers and more abstract patterns in the higher layers of the networkm, allows us to completely remove the hand-designed feature extracting process and treat CNNs as end-to-end learners. Each layer in the network uses the output of previous layes as “building blocks”to construct increasingly more abstract concepts. There layers are learned automatically – there is no hand-crafted feature engineering taking place in out network.

神经网络的历史：McCulloch and Pitts (binary classifier) -> Rosenblatt (Perceptron algorithm) -> Rumelhart (backpropagation algorithm) -> LeCun (Convolutional Neural Network)

Filter in lower levels of the network represent edges and corners, while higher level layers use the edges and corners to learn more abstract concepts useful for discriminating between image classes.

Histogram of Oriented Gradients (HOG)用于detecting objects in images.

CNN：instead of hand-defining a set of rules and algorithms to extract features from an image, there features are instead automatically learned from the training process.

the goal of machine learning: computers should be able to learn from experience of the problem they are trying to solve.

There is no consensus amongst experts on exactly what makes a neural network “deep”; however, we know that:

1. Deep learning algorithms learn in a hierarchical fashion and therefore stack multiple layers on top of each other to learn increasingly more abstract concepts.
2. A network should have >2 layers to be considered “deep”
3. A network with >10 layers is considered very deep

3、Image Fundamentals

The origin point (0,0) corresponds to the **upper-left** corner of the images. As we move down and to the right, both the x and y values increase, where we go x columns to the right and y rows down.

OpenCV and scikit-image represent RGB images as multi-dimensional NumPy arrays with shape (height, width, depth).

|  |
| --- |
| 1. **import** cv2 2. img = cv2.imread('example.jpg') 3. **print**(img.shape)  # 图像尺寸为1024\*768，而shape为(768,1024,3) 4. cv2.imshow("Image", img) 5. cv2.waitKey(0) 6. (b, g, r) = img[20, 100] |

aspect ratio: the ratio of the width to the height of the image.

Ignoring the aspect ratio can lead to images that look compressed and distorted. To prevent this behavior, we simple scale the width and height of an image by equal amounts when resizing an image.

Most neural networks and Convolutional Neural Networks applied to the task of image classification assume a fixed size input, meaning that the dimensions of all images you pass through the network must be the same. Common choices for width and height image sizes inputted to Convolutional Neural Networks include 32\*32, 64\*64, 224\*224, 227\*227, 256\*256, and 299\*299.

How are we supposed to preprocess these images?

1. simply ignore the aspect ratio and deal with the distortion
2. resizing the image along its shortest dimension and then taking the center crop.

4、Image Classification Basics

The goal of an image classification system is to take an input image and assign a label to it from our categories set.

In the context of image classification, our dataset is a collection of images. Each image is, therefore, a data point.

have to handle factors of variation: viewpoint variation, scale variation, deformation, occlusion variation, illumination, background clutter (背景杂波), intra-class variation.

canonical adj. 权威的

always consider the scope of your image classifier.

Instead of trying to construct a rule-based system to describe what each category “looks like”, we can instead take a data driven approach by supplying examples of what each category looks like and then teach our algorithm to recognize the difference between the categories using these examples.

four steps to constructing a deep learning model:

1. Gather your dataset

the number of images for each category should be approximately uniform.

Class imbalance

1. Split your dataset: 1) A training set; 2) A testing set

the training set and testing set are independent of each other and do not overlap. 一般分割比例为：66.6%/33.3%, 75%/25%, 90%/10%.

But what if you have hyperparameters to tune? normally allocate roughly 10~20% of the training data for validation.

1. Train your network
2. Evalute: precision, recall, f-measure

Generalization: the ability for a network to generalize and correctly predict the class label of an image that does not exist as part of its training or testing data.

如果模型的准确率欠佳：consider the set of factors of variation mentioned above. Does your training dataset accurately reflect exmplaes of thes factors of variation? If not, you will need to gather more training data.

5、Datasets for Image Classifiction

CALTECH-101：popular benchmark dataset for object detection.

6、Configuring your development environment

consider using a Linux environment such as Ubuntu.

7、Your first Image classification

should always be cognizant of your dataset size before even starting to work with image classification algorithm.

preprocessing methods for booosting classification accuracy: mean subtraction, sampleing random patches, simply resizing the image to a fixed size.

* a basic image preprocessor

对于分类图片，使用hierarchical directory structure保存，如all images inside the dog subdirectory are examples of dog. Directory structure为：/dataset\_name/class/image.jpg.

**k-NN**：directly relies on the distance between feature vectors, classifies unknown data points by finding the most common class among the k closest examples. Each data points in the k closest data points casts a vote, and the category with the highest number of votes wins.

First, we need to select a distance metric or similarity function:

Euclidean distance (L2 distance):

Manhattan/city block (L1 distance):

Many machine learning algorithms assume that the class labels are encoded as integers.

|  |
| --- |
| 1. **from** sklearn.preprocessing **import** LabelEncoder 2. le = LabelEncoder() 3. labels = le.fit\_transform(labels) |

The k-NN algorithm is unable to learn any discrimination patterns between these species. This is one of the primary drawbacks of the k-NN algorithm.

The k-NN algorithm is more suited for low-dimensional feature spaces. Distances in high-dimensional feature spaces are often unintuitive. Most importantly, it gives us a baseline that we can use to compare neural networks and Convolutional Neural Networks to as we progress through the rest of the book.

Drawbacks of k-NN：

1. it does not actually learn anything – if the algorithm makes a mistake, it has no way to correct and improve itself for later classification.
2. without specialized data structures, the k-NN algorithm scales linearly with the number of data points, making it not only practically challenging to use in high dimensions, but theoretically questionable in terms of its usage.

8、Parameterized learning

parametric model: a learning model that summarizes data with a set of parameters of fixed size. No matter how much data you throw at the parametetric model, it will not change its mind about how many parameters it needs.

Parameterization involves defining a problem in terms of four key components:

1. data: includes both the data points and their associated class labels.
2. a scoring function: accpets our data as an input and maps the data to class labels.
3. a loss function: quantifies how well our predicted class labels agree with our ground-truth labels.
4. weights and biases: that we will actually be optimizing. Based on the output of our scoring function and loss function, we will be tweaking and fiddling with the values of the weights and biases to increase classification accuracy.

a simple linear mapping:

其中，有K个unique categories，W的维度为，的维度为，b的维度为.

two primary advantages to utilizing parameterized learning:

1. Once we are done training our model, we can discard the input data and keep only the weight W and the bias vector b.
2. Classifying new test data is fast.

discuss two important concepts: 1) loss function; 2) optimization methods.

How we go about updating the parameters of weight matrix W or bias vector b is an optimization problem.

A loss function can be used to quantify how well our scoring function is doing at classifying input data points.

abbreviate our scoring function as s:

Which implies that we can obtain the predicted score of the j-th class via the i-th data point:

We can put it all together, obtaining the hinge loss function (multi-class SVM loss):

The hinge loss function is summing across all incorrect classes () and comparing the output of our scoring function s returned for the j-th class label (the incorrect class) and the class (the correct class). A given is classified correctly when the loss .

To derive the loss across our entire training set, we simply take the mean over each individual :

对于Image #1，Dog，Cat，Panda对应的score分别为4.26, 1.33, -1.01，其hinge loss为：

squared hinge loss:

Softmax classifiers give you probabilities for each class label while hinge loss gives you the margin.

rank-5 accuracy: check to see if the ground-truth label is in the top-5 predicted labels returned by a network for a given input image.

our loss function should minimize the negative log likelihood of the correct class:

其中，

所以，interpret these scores as unnormalized log probabilities for each class label, 即cross-entropy loss is:

9、Optimization methods and regularization

optimization algorithm: iteratively evaluate your parameters, compute your loss, then take a small step in the direction that will minimize your loss.

Compute the gradient W across all dimensions using the following equation:

In >1 dimensions, our gradient becomes a vector of partial derivatives. 该求梯度方法存在的问题为：1) it is an approximation to the gradient; 2) it is painfully slow.

What gradient descent is? Attempting to optimize our parameters for low loss and high classification accuracy via an iterative process of taking a step in the direction that minimizes loss.

Optimization algorithm may not be guaranteed to arrive at even a local minimum in a reasonable amount of time, but it often finds a very low value of the function quickly enough to be useful. – Goodfellow

bias trick: a method of combining our weight matrix W and bias vector b into a single parameter. To combine both the bias and weigh matrix, we add an extra dimension to our input data X that holds a constant 1 – this is our bias dimension. Allowing us to learn only a single matrix of weights.

Good initialization is critical to training a neural network in a reasonable amount of time, so random initialization along with sinple heuristics win out in the vast majority of circumstances.

Vanilla gradient descent only performs a weight update once for every epoch.

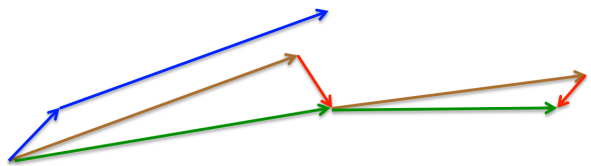
Stochastic Gradient Descent performs a weight update for every batch of training data, implying there are multiple weight updates per epoch. This approach leads to a faster, more stable convergence.

Typical batch sizes include 32, 64, 128, and 256.

Momentum: a method used to accelerate SGD, enabling it to learn faster by focusing on dimensions whose gradient point in the same direction.

The momentum term is commonly set to 0.9. Another common practice is to set to 0.5 until learning stabilizes and then increase it to 0.9.

Nesterov acceleration can be conceptualized as a corrective update to the momentum which lets us obtain an approximate idea of where our parameters will be after the update. Under Nesterov acceleration we would first make a big jump in the direction of our previous gradient (brown vector), measure the gradient, and then make a correction.



Rule of thumb: whenever using SGD, also apply momentum. In most cases, you can set it to 0.9. Although Karpathy suggests starting at 0.5 and increasing it to larger values as your epochs increase. SGD os easier to work with large datasets when using momentum, while, smaller datasets tend to enjoy the benefits of Nestetov acceleration.

Various types of regularization techniques: L1 regularization, L2 regularization (weight decay), Elastic Net, dropout, data augmentation, early stopping.

Regularization helps us control our model capacity.

A regularization penalty, a function that operates on our weight matrix, commonly written as a function, .

加入正则化后，损失函数变为：

Both the learning rate and the regularization term are the hyperparameters that you will spending the most time tuning.

Standard weight update rule:

Taking into account regularization, the weight uodate rule becomes:

Which regularization method you should use?

Treating this choice as a hyperparameter you need to optimize over and perform experiments to determine if regularization should be applied, and if so which method of regularization, and what the proper value of is.

Regularization can provide a boost in our testing accuracy and reduce overfitting, provided we can tune the hyperparameters right.

10、Neural network fundamentals

directed graph：有向图

Recommand starting with a ReLU to obtain a baseline accuracy, tune my network and optimizer parameters (architecture, learning rate, regularization strength, etc), and note the accuracy. Once reasonably satisfied with the acuracy, swap in an ELU.

Feedforward neural network: a connection between nodes is only allowed from nodes in layer to nodes in layer . There ar no backward or inter-layer connections allowed.

Perceptron training procedure:

1. Initialize our weight vector with small random values.
2. Until Perceptron converges:
   1. Loop over each feature vector and true class label in our training set D;
   2. Take and pass it through the network, calculating the output value: ;
   3. Update the weights : for all features .

The Perceptron training process is allowed to proceed until all training samples are classified correctly or a preset number of epochs is reached. it will never be able to correctly model the XOR function with a single layer Perceptron.

|  |
| --- |
| 1. **class** Perceptron: 2. **def** \_\_init\_\_(self, N, alpha=0.1): 3. # N: the number of columns in our input feature vectors 4. self.W = np.random.randn(N + 1) / np.sqrt(N) 5. self.alpha = alpha 7. **def** step(self, x): 8. **return** 1 **if** x > 0 **else** 0 10. **def** fit(self, X, y, epochs=10): 11. X = np.c\_[X, np.ones((X.shape[0]))] 12. **for** epoch **in** np.arange(0, epochs): 13. **for** (x, target) **in** zip(X, y): 14. p = self.step(np.dot(x, self.W)) 15. **if** p != target: 16. error = p - target 17. self.W += -self.alpha \* error \* x 19. **def** predict(self, X, addBias=True): 20. X = np.atleast\_2d(X) 21. **if** addBias: 22. X = np.c\_[X, np.ones((X.shape[0]))] 23. **return** self.step(np.dot(X, self.W)) |

The backpropagation algorithm phases:

1. The forward pass where our inputs are passed through the network and output predictions obtained.
2. The backward pass where we compute the gradient of the loss function at the final layer of the network and use this gradient to recursively apply the chain rule to update the weights in our network.