Deep Learning

Why Did We Arrive Here (Today)?

Introduction

1. Why such interest?
2. Why now?
3. Most famous example
4. In what conditions is this not overfitting?
5. In what conditions is this overfitting?

DataScience@SMU

Why Such Interest?

Why Such Interest?

- Terminology
  - Neural networks (NN) sound like brains
  - NN reintroduced the terminology of AI
- Disruptive technology
  - Replace human sensing
    - (OCR, voice recognition)
  - Replace human movement
    - (Robots, cars)
- Before, no models were plausible replacements for such tasks.

DataScience@SMU

Why Now?

Why Now?

Since NN are not new, why now
- GPUs
- Aggregation of massive datasets

Secondary
- Stagnating research on other topics
- Let CS engineers contribute

DataScience@SMU

Most Famous Example

Most Famous Example

Convolutional neural networks

- Vectorize an image
- Convolutional NN

Feature representation

- CNN
- GANS
- Autoencoders
- Recurrent neural networks
  LSTM

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
from keras import backend as K
batch_size = 128
num_classes = 10
epochs = 12
img_rows, img_cols = 28, 28
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
input_shape = (img_rows, img_cols, 1)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
```

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

DataScience@SMU

In What Conditions Is This Not Overfitting?

Why Is This Not Overfitting?

Accurate collection of distribution

- Claim: Google has a representative sample of all images (i.e., the distribution is stable and there is ample supply).
- Claim: Amazon has representative sample of product reviews (?).
- Claim: Newspapers have a reasonable estimate of how newspapers are written.
- We are using a holdout set that we only look at after we have decided what our optimal model is.
- We have ample enough data to retrain our model whenever we need to, using a new holdout set.
- We've evaluated core techniques for increasing generalizability such as dropout, etc.

DataScience@SMU

In What Conditions Is This Overfitting?

Why Is This Overfitting?

- We never have a stable distribution.
- We never have enough data (too few or too hard to access).
- Engineers often cut corners in model building.
- Companies gain an edge by working at the edge and the stability we require for DL is not there.
- Deep learning memorizes relationships within the distribution but does nothing to generalize this knowledge.
- Malicious attacks are not yet included in the training data!
- When distributions change, how will we know?

DataScience@SMU