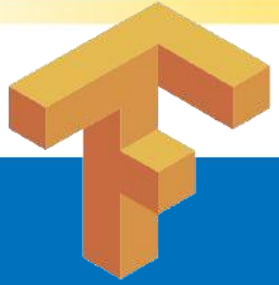


TensorFlow Ops

Tensor Flow for Deep Learning Research

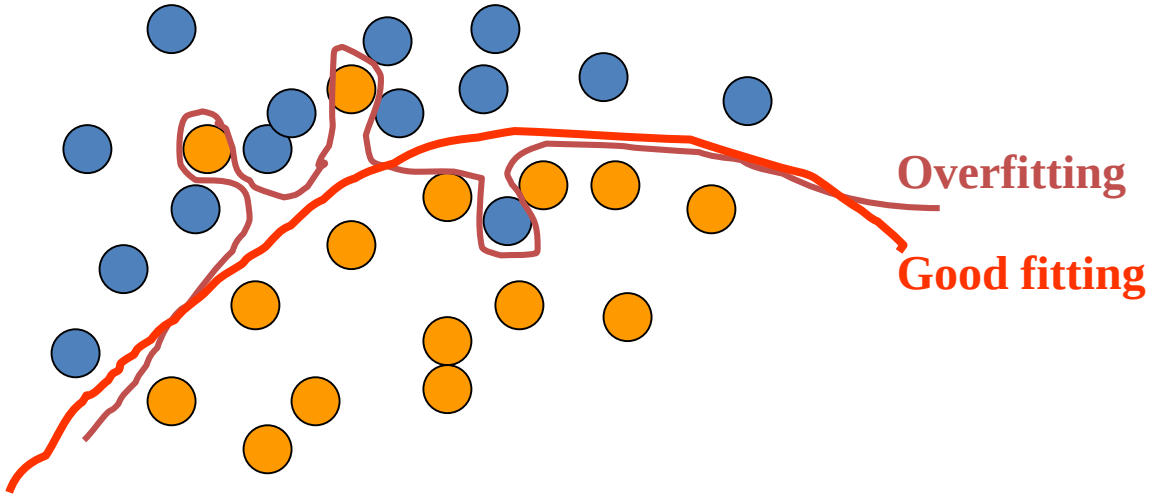
UMKC

Feedback is greatly appreciated!

Overview

- Overfitting
- Dropout
- Data augmentation
- Callbacks
- Word Embeddings (representation of text data)

Overfitting and Good Fitting



Overfitting function can not generalize well to unseen data.

UMKC

Preventing Overfitting

Standard ways to limit the capacity of a neural net:

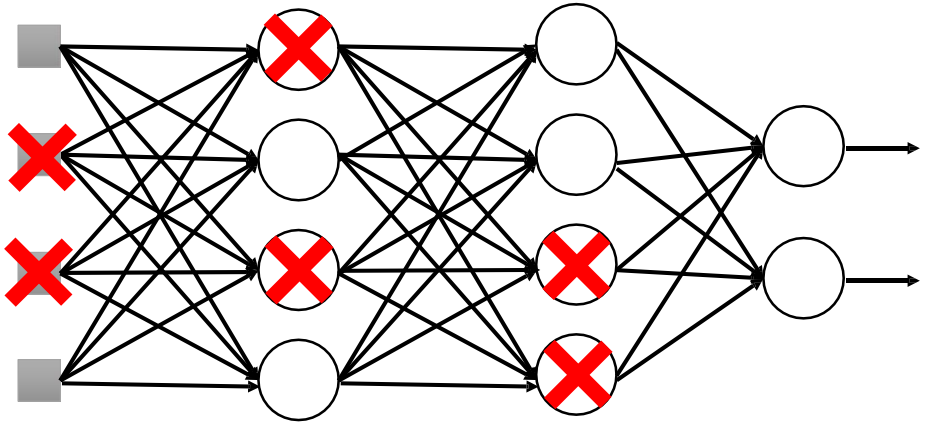
- Limit the number of hidden units.
- Limit the size of the weights.
- Stop the learning before it has time to overfit.

G. Hinton, 2006

UMKC

Dropout

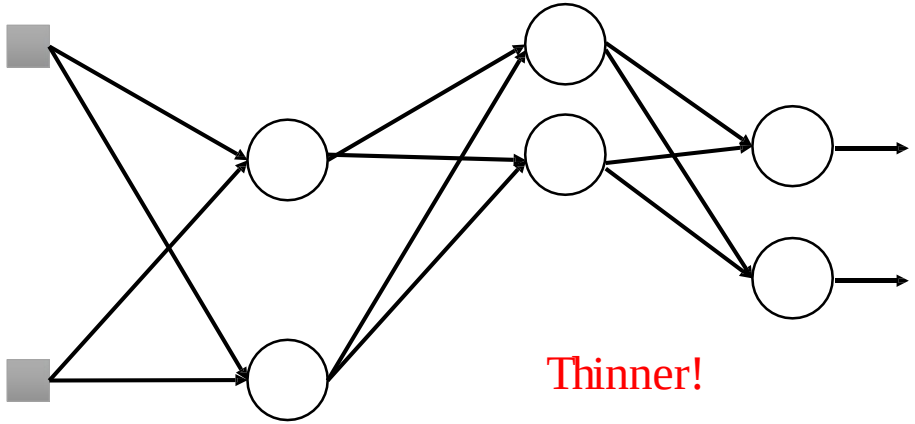
Training:



- **Each time before updating the parameters**
  - Each neuron has p% to dropout

Dropout

Training:

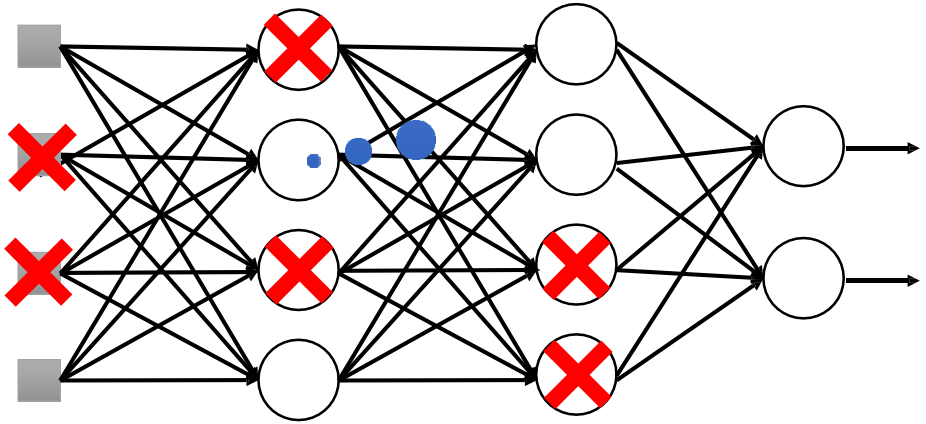


- **Each time before updating the parameters**
  - Each neuron has p% to dropout
    - **The structure of the network is changed.**
  - Using the new network for training

For each mini-batch, we resample the dropout neurons

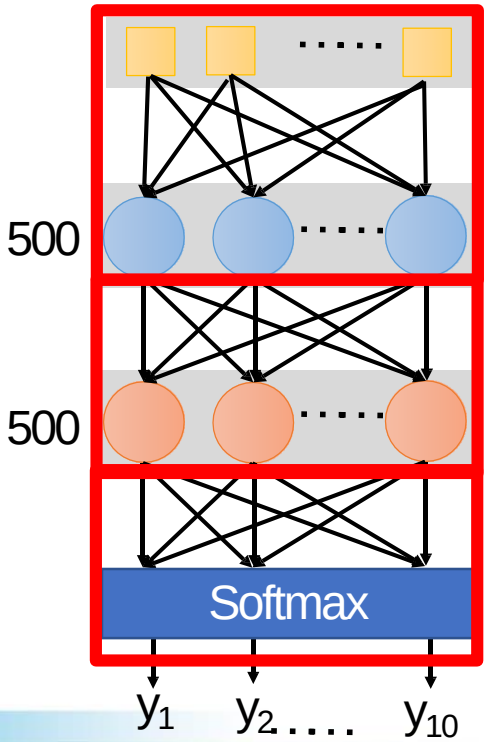
UMKC

Dropout - Intuitive Reason



- When teams up, if everyone expect the partner will do the work, nothing will be done finally.
- However, if you know your partner will dropout, you will do better.

Let's try it




```
model = Sequential()
```

```
model.add( Dense( input_dim=28*28,
                  output_dim=500 ))
model.add( Activation('sigmoid') )
```

model.add(dropout(0.8))

```
model.add( Dense( output_dim=500 ) )
model.add( Activation('sigmoid') )
```

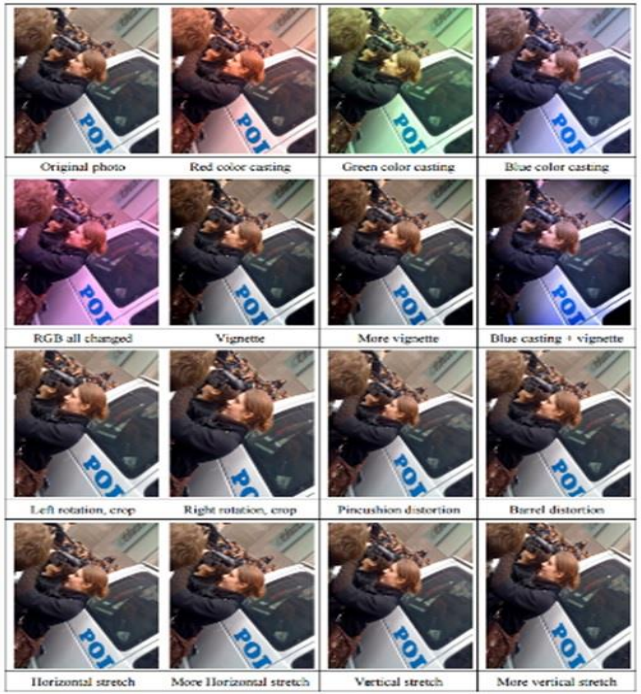
model.add(dropout(0.8))

```
model.add( Dense(output_dim=10 ) )
model.add( Activation('softmax') )
```

UMKC

Data Augmentation (Jittering)

- Create *virtual* training samples
  - Horizontal flip
  - Random crop
  - Color casting
  - Geometric distortion



Deep Image [Wu et al. 2015]

UMKC

Programming

callbacks

- A callback is a set of functions to be applied at given stages of the training procedure
- You can use callbacks to get a view on internal states and statistics of the model during training
- keras.callbacks.Callback()

callbacks

- **History**
- **ModelCheckpoint**
- **EarlyStopping**
- **TensorBoard**
- **ReduceLROnPlateau**

UMKC

History

keras.callbacks.History()

ModelCheckpoint

```
keras.callbacks.ModelCheckpoint(filepath, monitor='val_loss', verbose=0,
save_best_only=False, save_weights_only=False, mode='auto')
```

Early Stopping

```
keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0,
patience=0, verbose=0, mode='auto', baseline=None,
restore_best_weights=False)
```

ReduceLROnPlateau

```
keras.callbacks.ReduceLROnPlateau(monitor='val_loss', patience=10,
verbose=0, mode='auto', min_delta=0.0001, min_lr=0)
```

Tensorboard

- Tensorflow, the deep learning framework from Google comes with a great tool to debug
- It hosts a website on your local machine in which you can monitor things like accuracy, cost functions
- and visualize the computational graph that Tensorflow is running based on what you defined in Keras
- Pip install tensorboard

UMKC

Tensorboard

```
tensorboard = TensorBoard(log_dir="logs/{}".format(time())) model.fit(x_train, y_train, verbose=1, callbacks=[tensorboard])
```

- You need to create a new TensorBoard instance and point it to a log directory where data should be collected.
- Next you need to modify the fit call so that it includes the tensorboard callback.

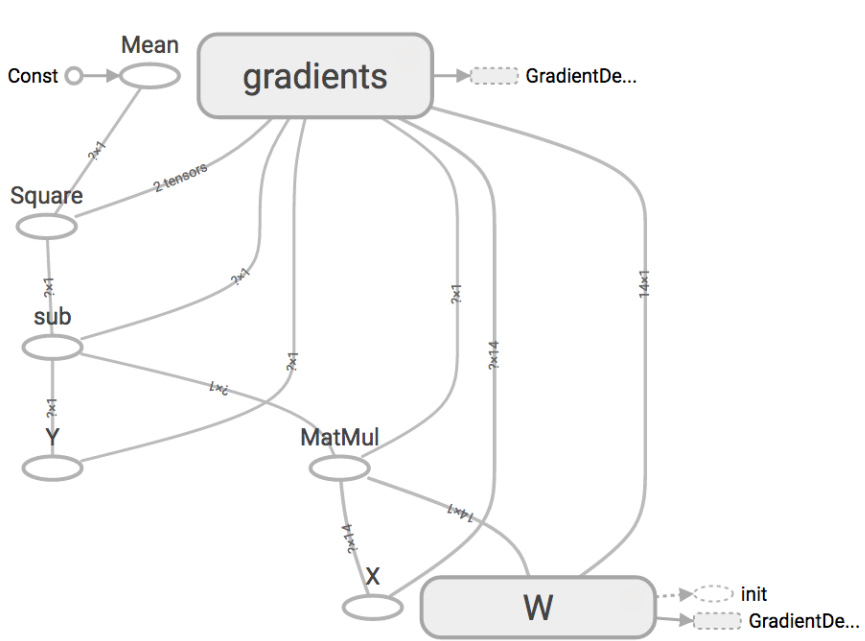
Then the following command in the separate terminal

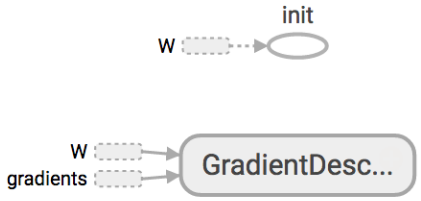
```
tensorboard --logdir=logs/
```

http://localhost:6006/

UMKC

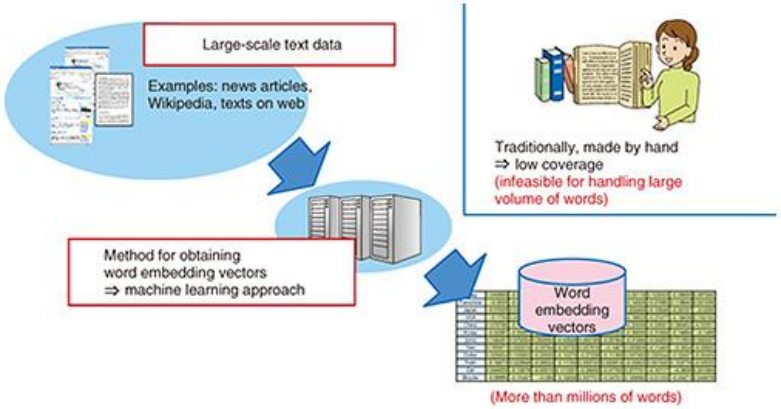
Tensorboard





Why Word Embeddings: Essential for text processing with DL

- It is an essential part of text processing with Deep Learning because Machine Learning algorithms, and almost all Deep Learning architectures, are incapable of processing strings or plain text in their raw form.



TEXT processing

Source: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201711fa4.html

UMKC

Types of Word Embeddings

Word embeddings can be broadly classified into two categories

- Frequency based Embedding
  - Count Vector
  - TF-IDF Vector
  - Co-Occurrence Vector
- Prediction based Embedding
  - word2vec
    - CBOW (Continuous Bag of words)
    - Skip-Gram Model
  - GloVe: Global Vectors for Word Representation

Source: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

UMKC

References

https://deeplearning4j.org/word2vec.html
https://towardsdatascience.com/tagged/word2vec
https://medium.com/@Aj.Cheng/word2vec-3b2cc79d674
https://medium.com/tag/word2vec
https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
https://becominghuman.ai/how-does-word2vecs-skip-gram-work-f92e0525def4

UMKC