# DL Dev Course: Week 02

RED DRAGON AI

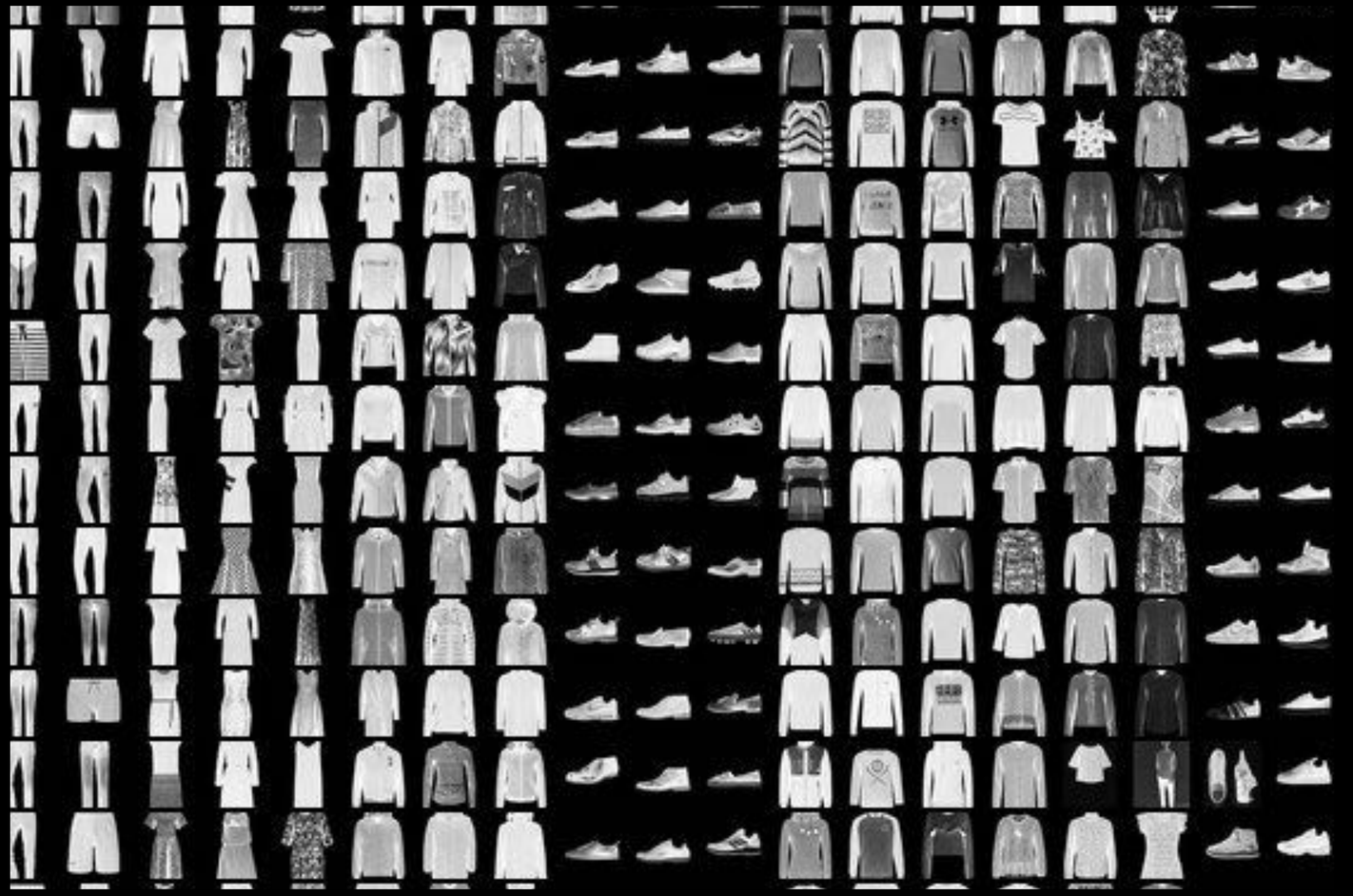
# Uses of CNNs

- Classification
- Object Detection
- Style Transfer
- Image Segmentation
- Generative Model - Super Resolution etc.
- Text models

# MNIST

**Original MNIST**

10 classes

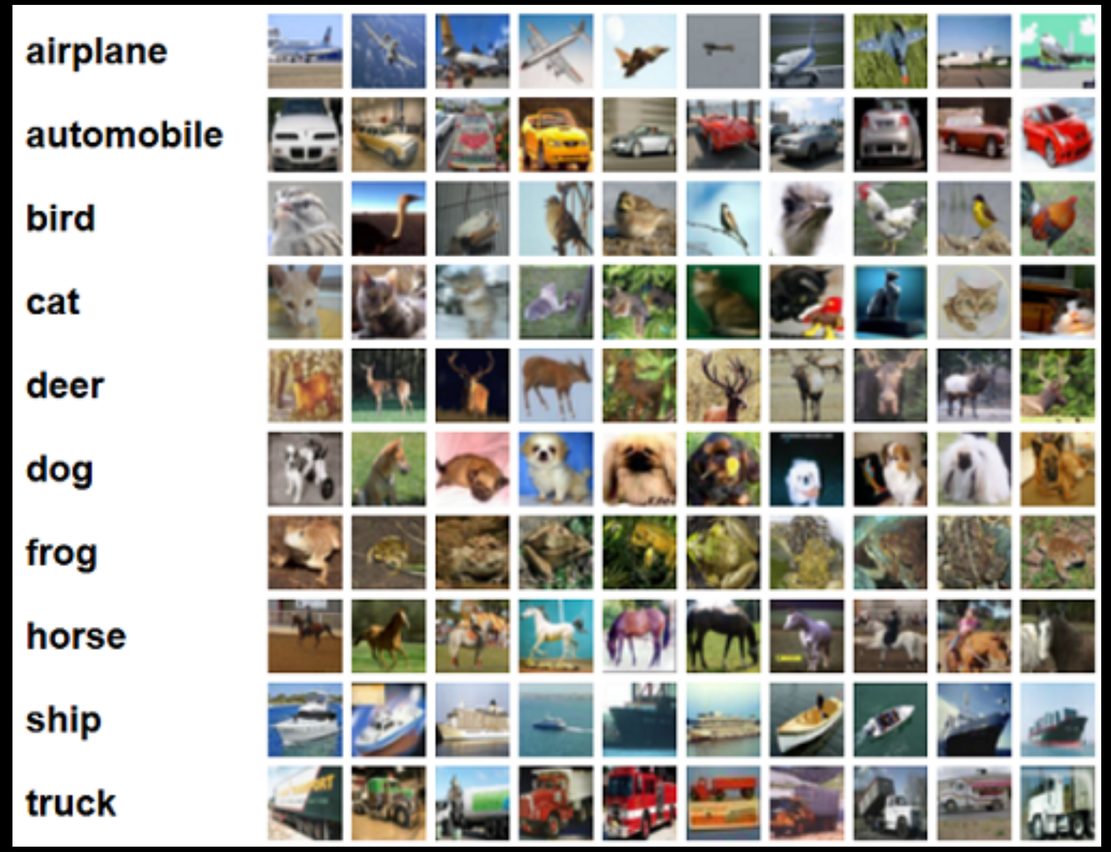


**Fashion MNIST**

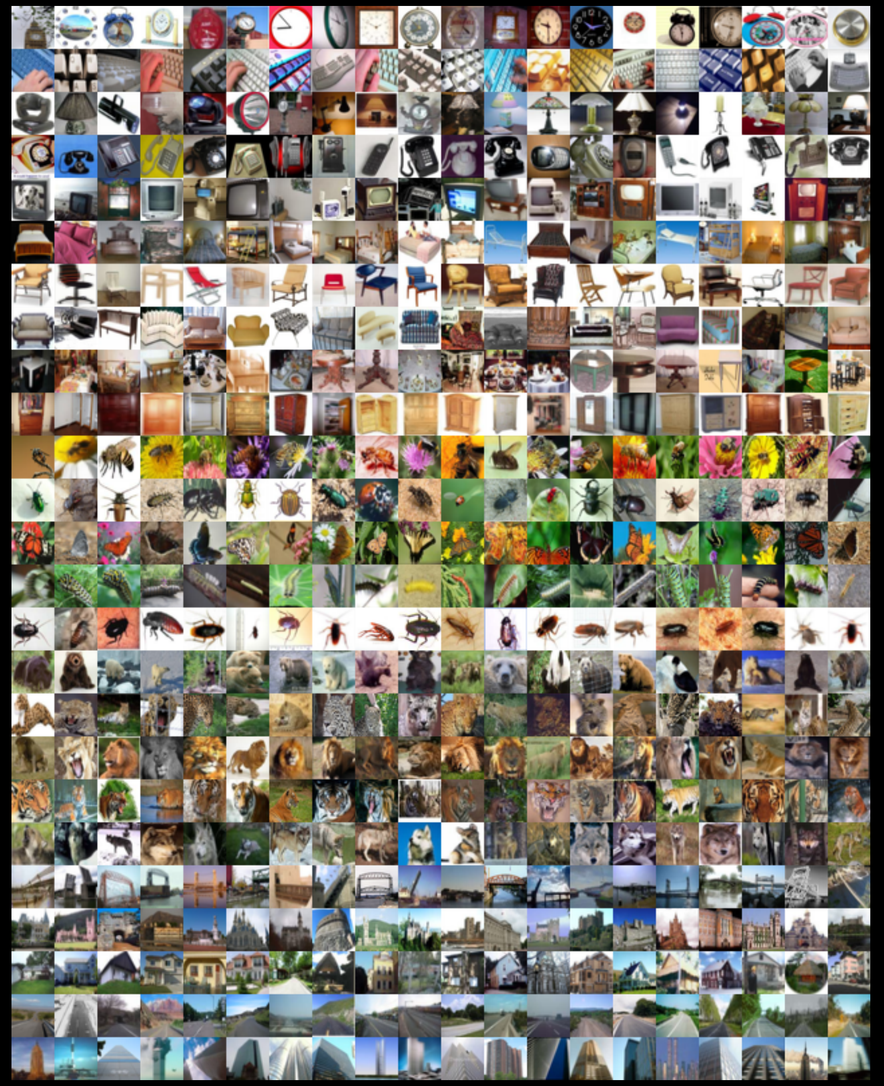
10 classes

RED DRAGON AI

# CIFAR




**CIFAR10**
10 classes



**CIFAR100**
100 classes

RED DRAGON AI

# IMAGENET

## ImageNet Dataset


Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*. [web]

3

**1000 classes**

RED DRAGON AI

# CNNs Timeline

**MNIST Dataset :**
Subset of NIST Dataset 1980-90s
http://yann.lecun.com/exdb/mnist/

**LeNet 1-5 : Yann LeCun**
1998 - 0.7-0.8 Test Error Rate on MNIST
http://yann.lecun.com/exdb/publis/index.html

**CiFAR10 :**
2009
https://www.cs.toronto.edu/~kriz/cifar.html

RED DRAGON AI

# CNNs Timeline

**LeNet 1-5 : Yann LeCun**
**1998 - 0.7-0.8 Test Error Rate on MNIST**
**http://yann.lecun.com/exdb/publis/index.html**

**CiFAR10 :**
**2009**
**https://www.cs.toronto.edu/~kriz/cifar.html**

**GoogLeNet :**
**2014**
**https://arxiv.org/abs/1409.4842**

RED DRAGON AI

# CNNs Timeline

**CiFAR10 :**
2009
https://www.cs.toronto.edu/~kriz/cifar.html

**GoogLeNet :**
2014
https://arxiv.org/abs/1409.4842

**VGG 16/19 :**
**2014 - Visual Geometry Group**
**https://arxiv.org/pdf/1409.1556**

RED DRAGON AI

# CNNs Timeline

**GoogLeNet :**
**2014**
**https://arxiv.org/abs/1409.4842**

**VGG 16/19 :**
**2014 - Visual Geometry Group**
**https://arxiv.org/pdf/1409.1556**

**Resnet:**
**2015 - Microsoft**
**https://arxiv.org/pdf/1409.1556**

RED DRAGON AI

# CNNs Timeline

**VGG 16/19 :**
**2014 - Visual Geometry Group**
**https://arxiv.org/pdf/1409.1556**

**Resnet:**
**2015 - Microsoft**
**https://arxiv.org/pdf/1409.1556**

**InceptionV3:**
**2015 - Google**

RED DRAGON AI

# CNNs Timeline

**Resnet:**
2015 - Microsoft
https://arxiv.org/pdf/1409.1556

**InceptionV3:**
2015 - Google

**InceptionV4:**
2016 Feb - Google
https://arxiv.org/abs/1602.07261

RED DRAGON AI

# CNNs Timeline

**InceptionV3:**
**2015 - Google**

**InceptionV4:**
**2016 Feb - Google**
**https://arxiv.org/abs/1602.07261**

**InceptionV5,6,7:**
**2016 - Google**

RED DRAGON AI

# CNNs Timeline

**InceptionV4:**
**2016 Feb - Google**
**https://arxiv.org/abs/1602.07261**

**InceptionV5,6,7:**
**2016 - Google**

**Densenets**

**U-nets**

RED DRAGON AI

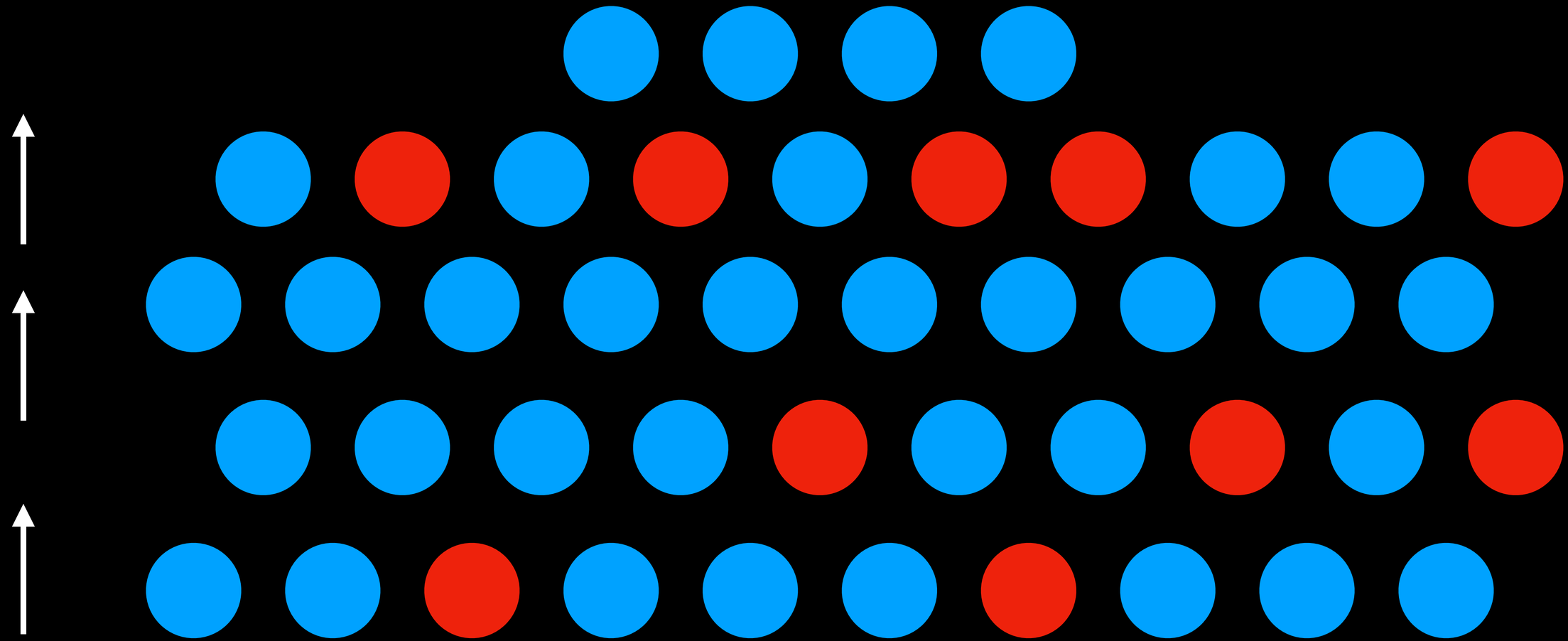
# VGG 16/19

- Very popular on Kaggle and for Transfer Learning tasks
- Because its simple to pull apart and you can chop and take from almost every level
- One of the first CNN networks to use Dropout

# Dropout

- Take X % of neurons and turn them off
- In Keras Dropout(percentage)
- The percentage is the percentage of that layer you are throwing away
- In some frameworks it is stated as percentage you are keeping

# Dropout



# Dropout

- It prevents the network from just memorising the data.
- By throwing away bits of learning in the network randomly, you prevent the network from learning to overfit the data.
- If you drop out in an early layer, you're losing that information for all of the future layers.  Be careful not to dropout too much in the early layers.  You can always add more dropout more in the later layers instead.
- Can be performed both on Dense layers and Convolution layers

# Underfitting

- The Model is not complex or powerful enough.
- Not enough parameters to get a good learning of the data distribution.
- Can be caused by too much dropout, not enough training data
- Prevent by :
  - Increasing features
  - Removing or reducing dropout

RED DRAGON AI

# Overfitting

- The Model is memorising the training set rather than generalising so that it can predict other examples it has never seen.
- The model is perhaps too complex  or it has learnt too much on the training data set
- Training dataset has a much higher accuracy than validation dataset

RED DRAGON AI

# Preventing Overfitting

- Prevent by:
  - 1. Adding more data
  - 2. Using Dropout
  - 3. Using image Augmentation
  - 4. Reduce network size
  - 5. Regularization

# Data Augmentation

- We make more images with the same data by creating variations
- Zoom
- Rotation
- Move left/right
- Flip
- Color changes

RED DRAGON AI

# Data Augmentation






RED DRAGON AI

# Batch Normalization

- Training can be 10X faster than not using it, because it often lets you use a 10X higher learning rate.
- It reduces overfitting without removing any information from the model.
- BatchNorm normalizes activations not just your inputs and adds trainable parameters that can control the scale of the activations overall.
- The model and gradients can rescale a whole set of weights for a layer.
- Always try to do it
- Older models don’t have it as its only been around for a couple of years.

# Mini Challenge 01

- QuickDraw dataset
- How many classes can you do?
- Accuracy?
- Network type
- Practice in training a bigger dataset

RED DRAGON AI