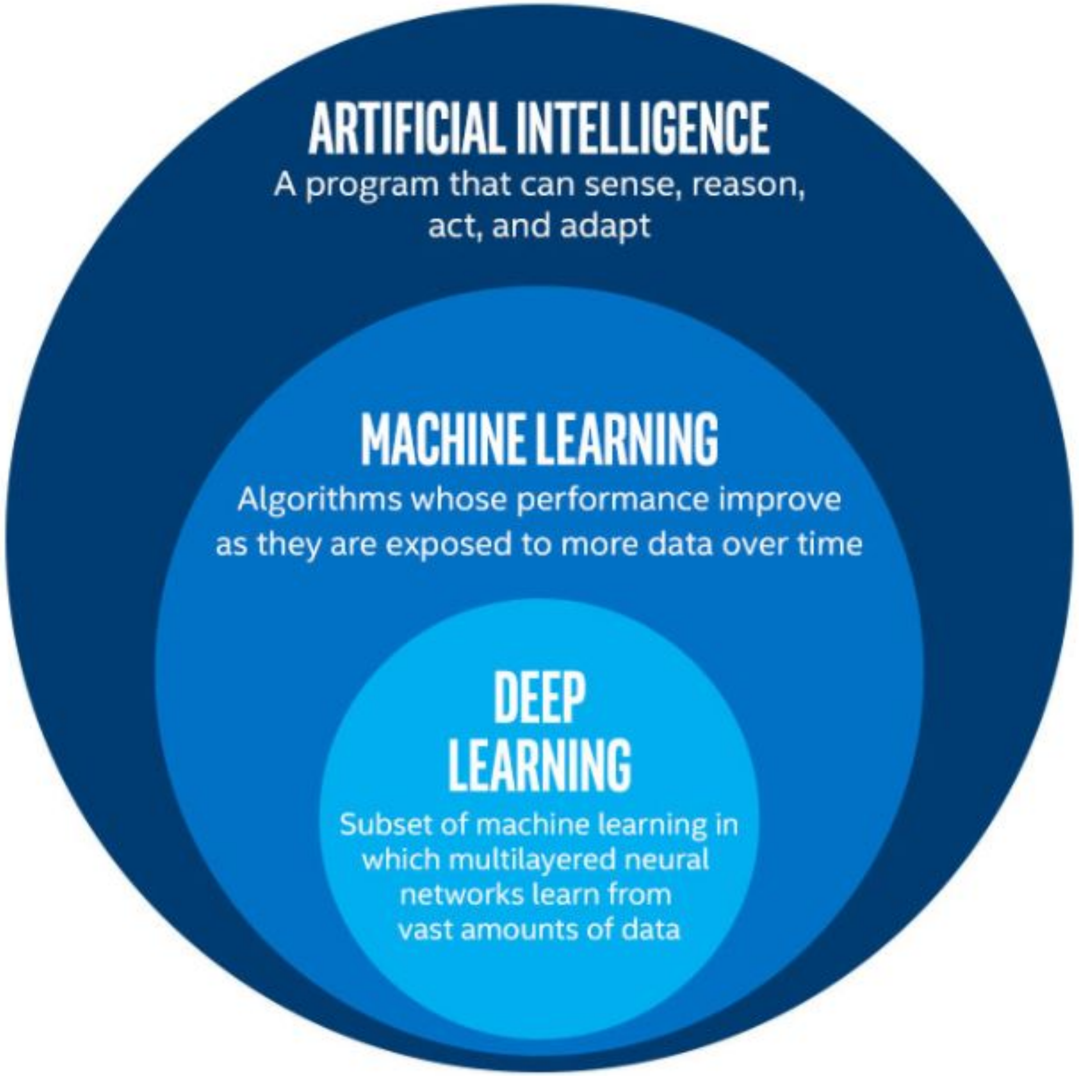
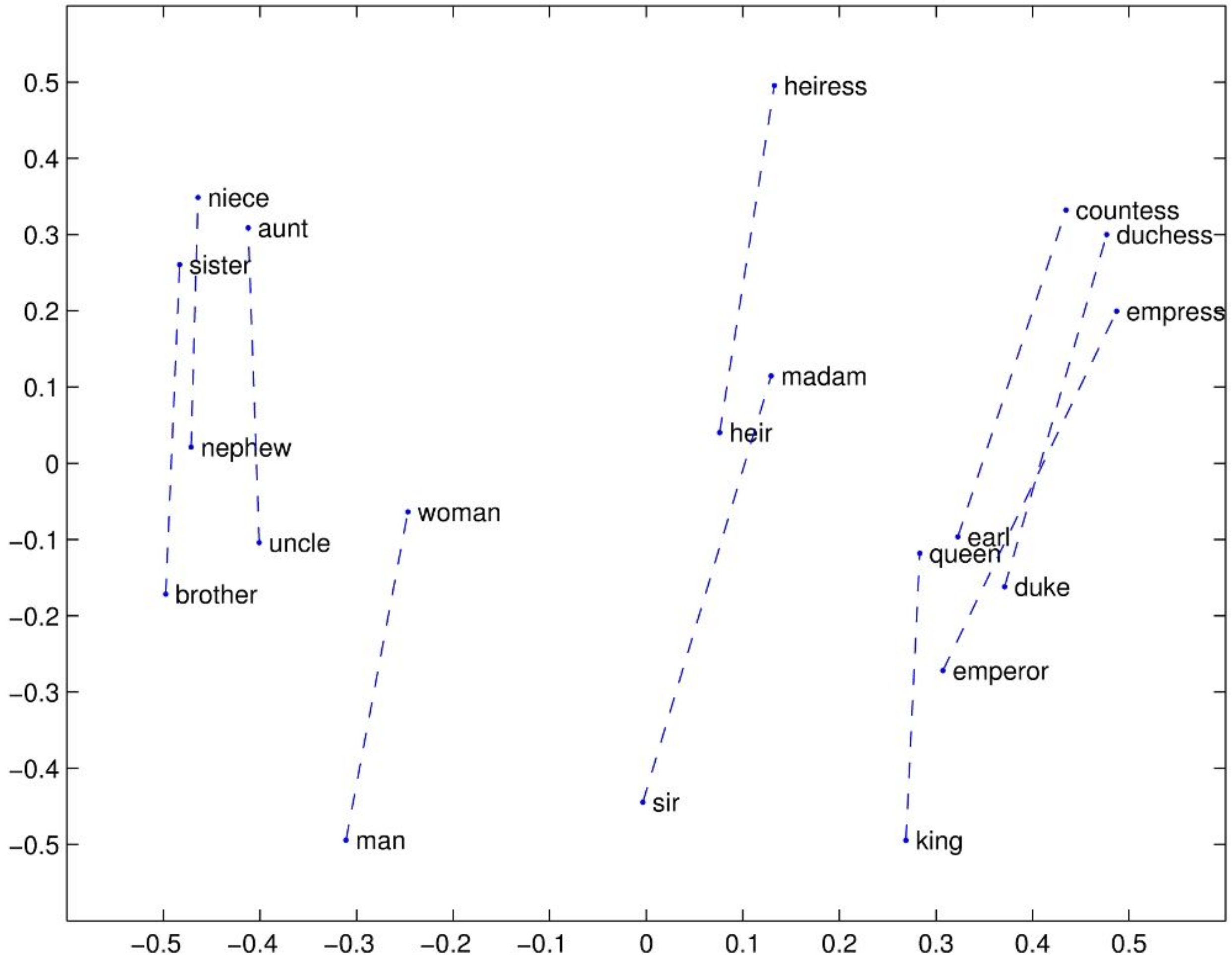
Data Science and Artificial Intelligence Introduction

What is Artificial Intelligence?



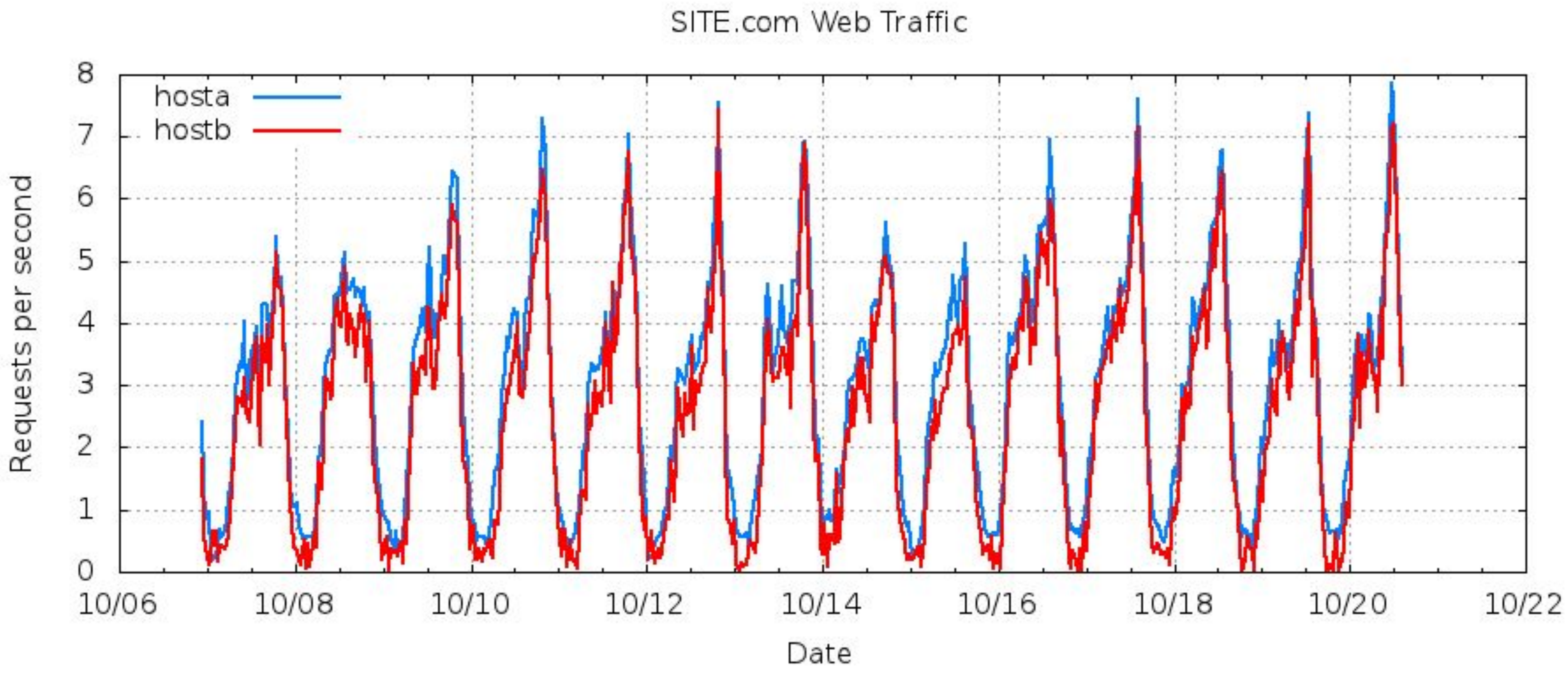
AI is not just machine learning, and machine learning is not just deep learning!

Different types of data: Text data



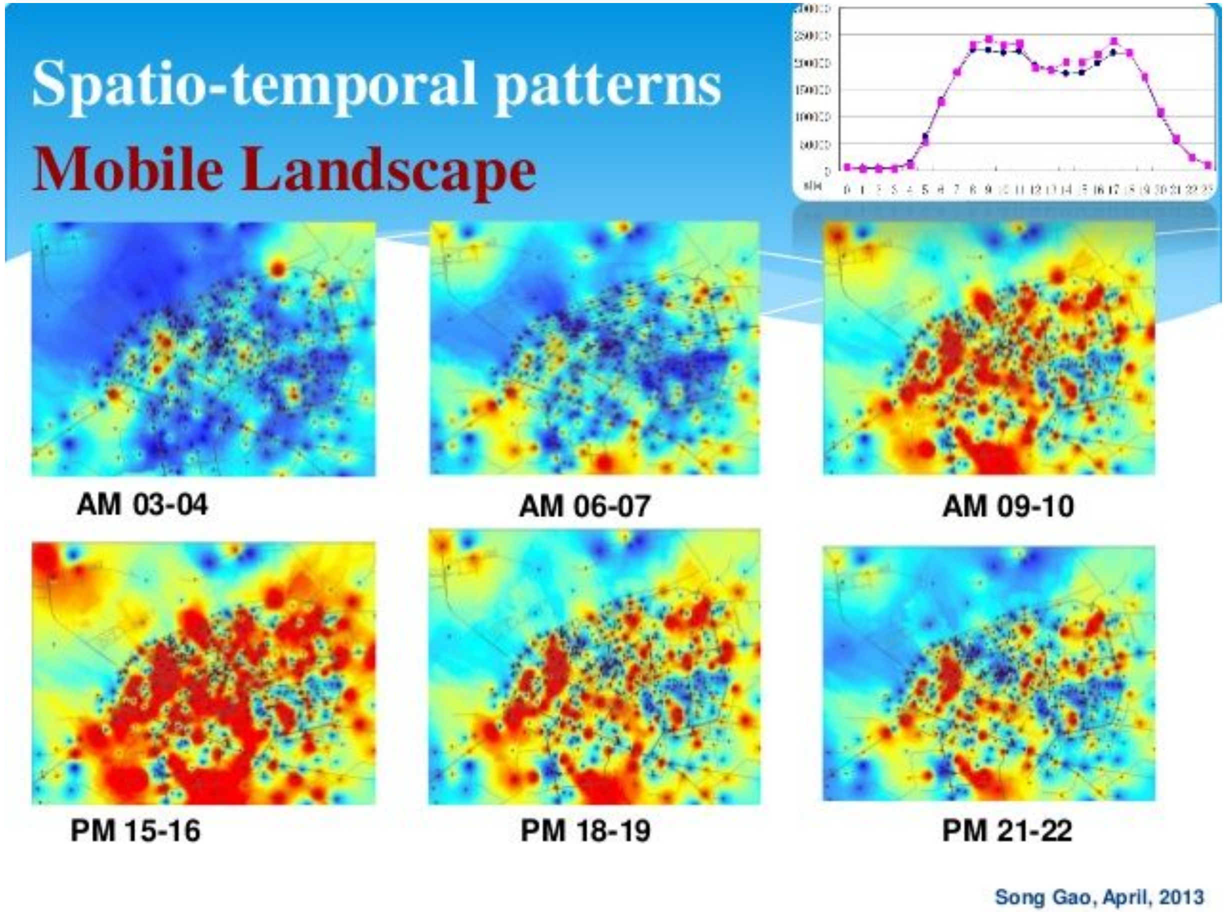
Source: https://nlp.stanford.edu/projects/glove/

Different types of data: Time Series Data

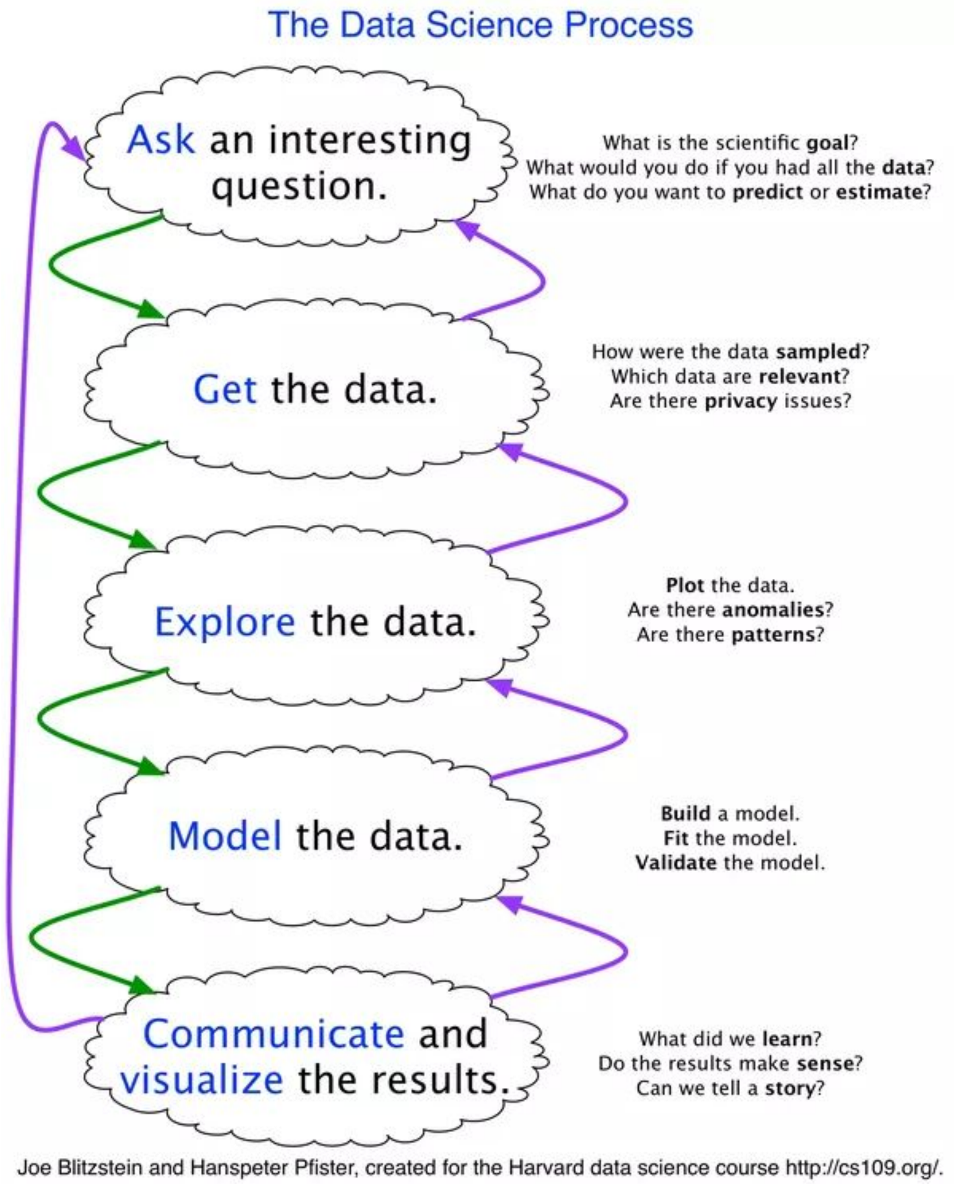


Different types of data: Image Data

Different types of data: Spatio-temporal Data

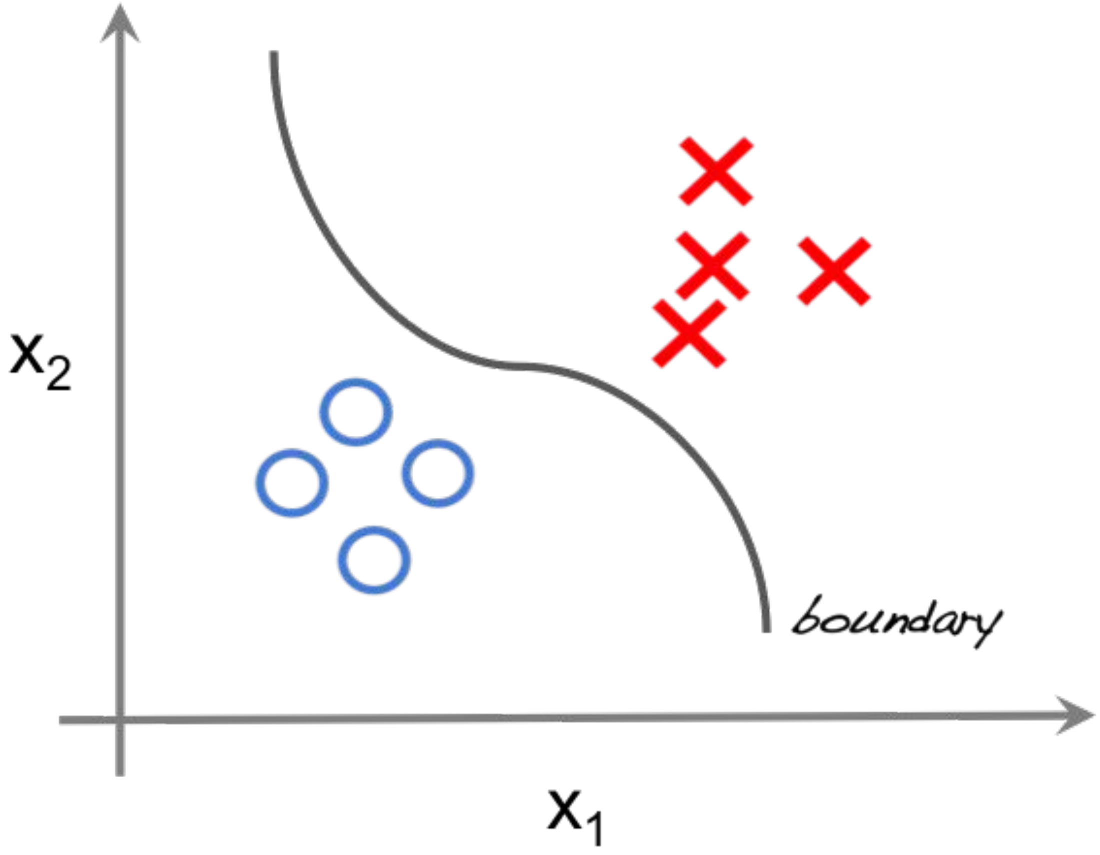


Data Science Workflow

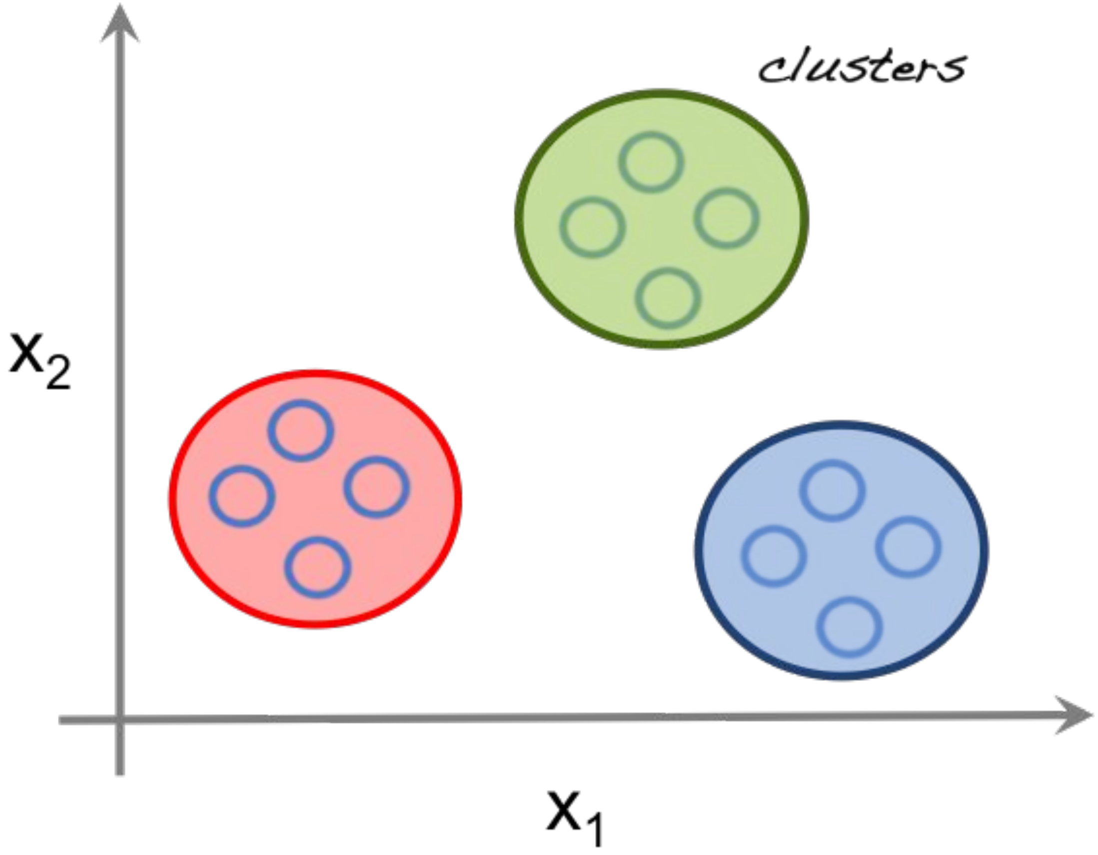


Supervised vs Unsupervised Learning

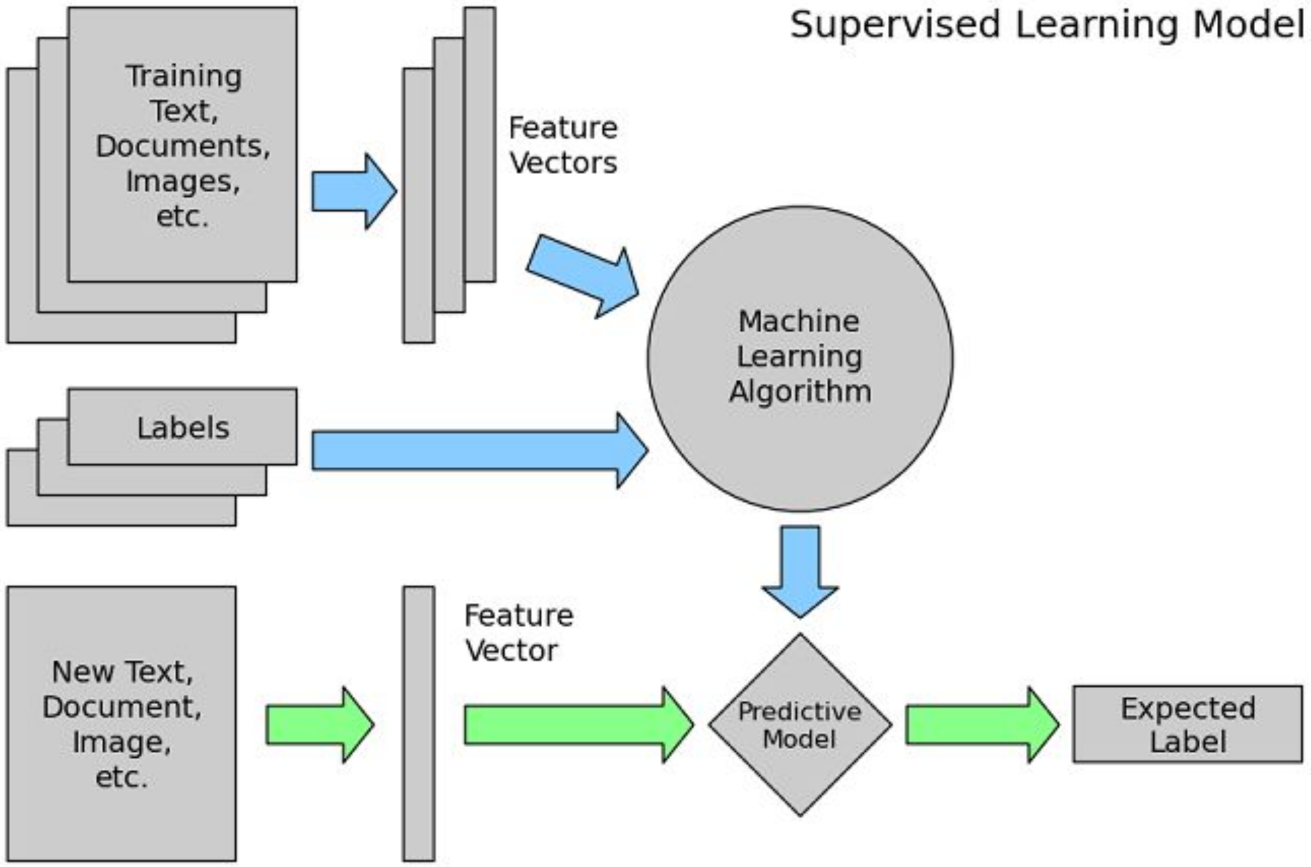
Supervised learning



Unsupervised learning



Supervised Learning Process



Working example – Simple Image Classification

- The MNIST database of handwritten digits (http://yann.lecun.com/exdb/mnist/) has a training set of 60,000 examples, and a test set of 10,000 examples.

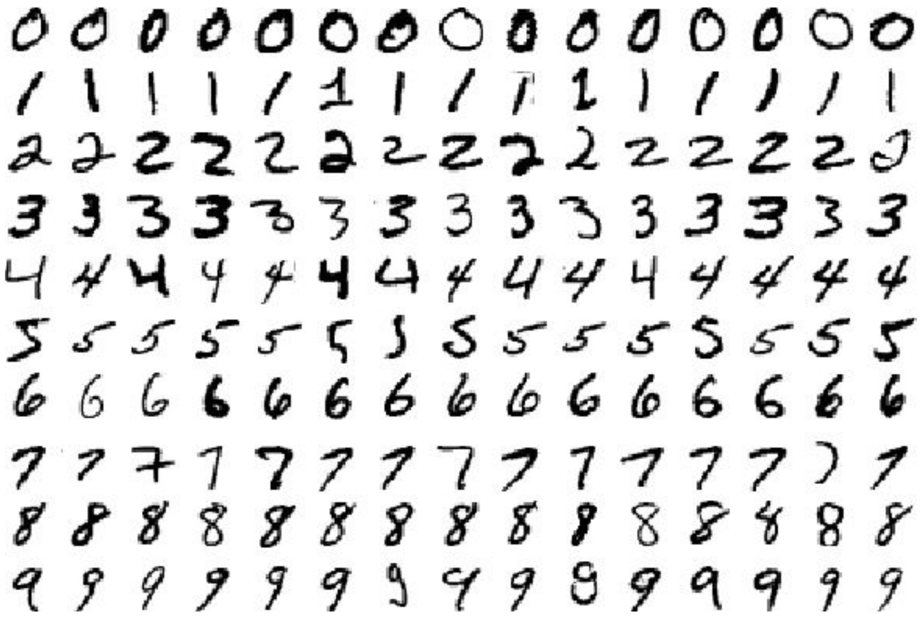
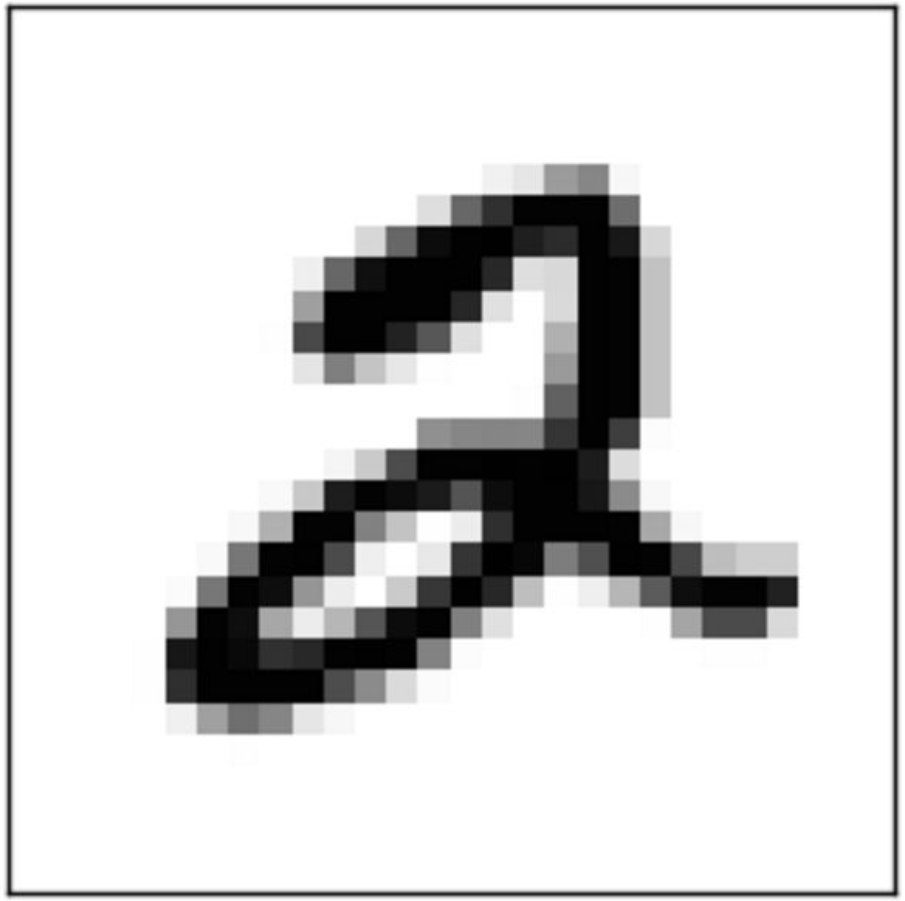


Image as input



Loss function in a softmax classifier

Cross-entropy loss

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \qquad \text{or equivalently} \qquad L_i = -f_{y_i} + \log\sum_j e^{f_j}$$

Softmax function

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

Minimizing the loss function – Gradient Descent

- Compute the best direction along which we should change our weight vector that is mathematically guaranteed to the direction of steepest descent
- This direction is based on the gradient of the loss function – we update the weights in the negative direction of the gradient, since we want to minimize the loss function

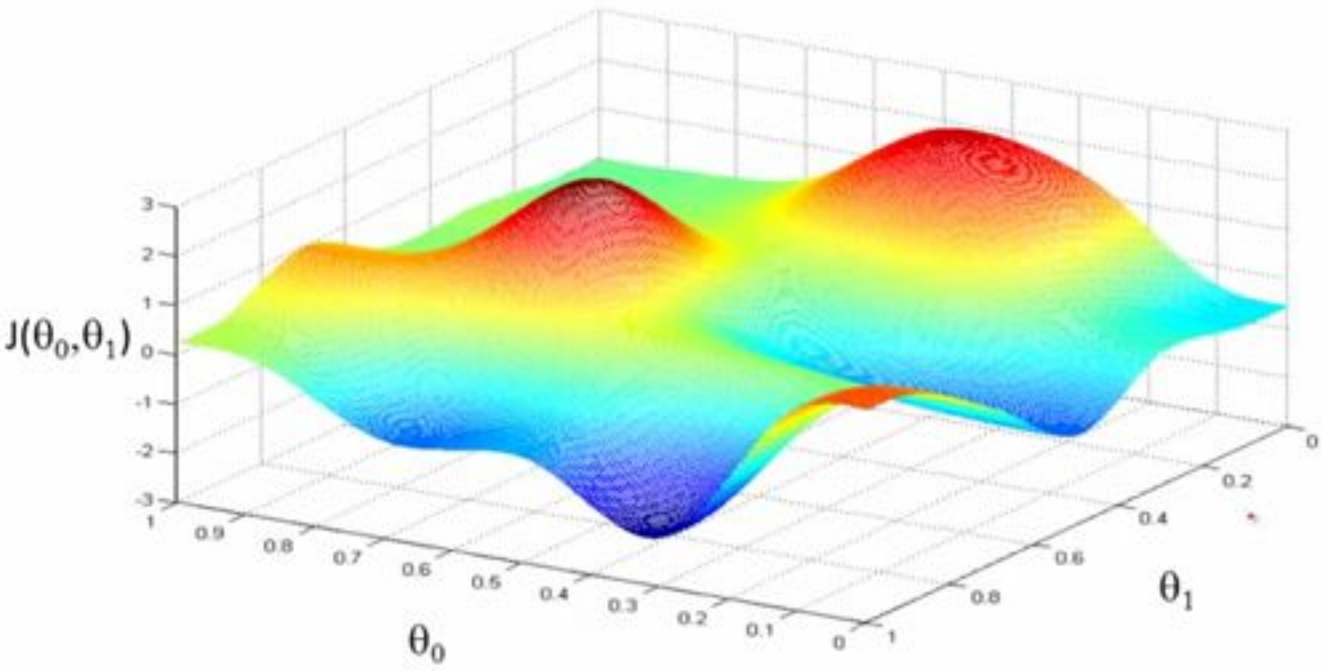
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Mini-batch gradient descent

- In most deep learning models, you have training data with millions of examples –> very slow to compute the loss function over all the training data, just to perform a single parameter update
- This term sometimes used interchangeably with Stochastic gradient descent (SGD), but to be precise, SGD refers to doing a parameter update with **each** training example

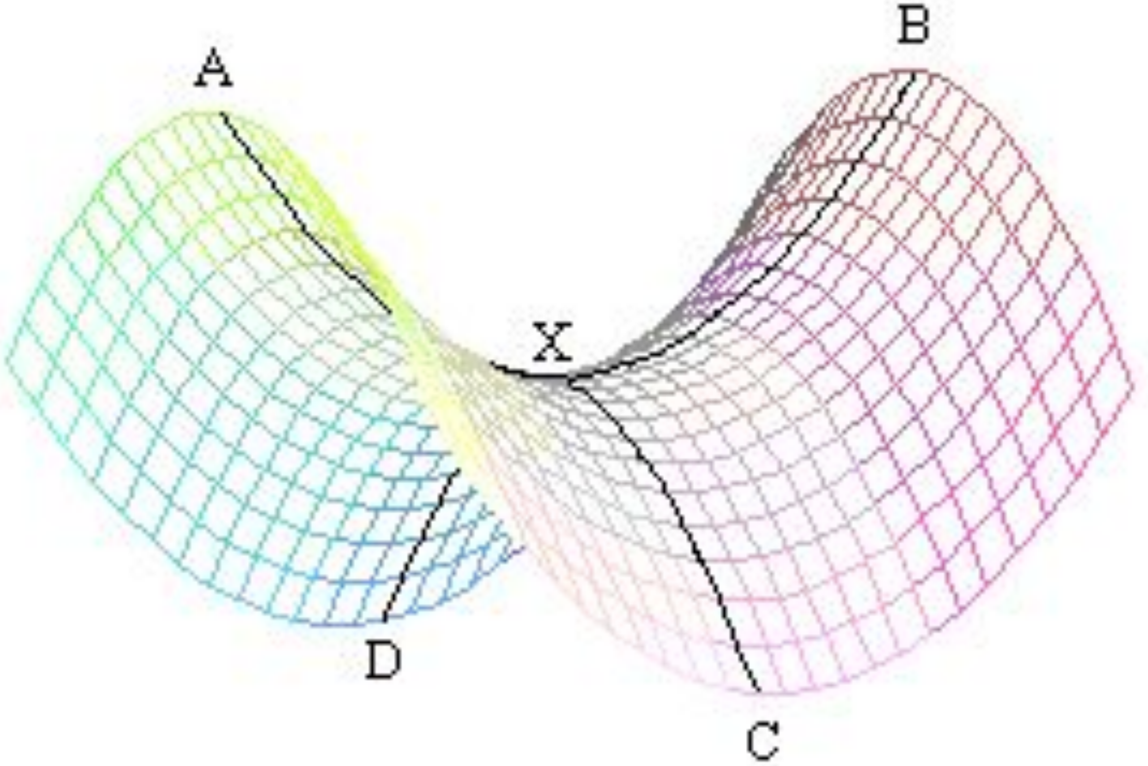
Problems faced with optimizing gradient descent

- Vanilla gradient descent is susceptible to local minima in non-convex functions



Source: Andrew Ng, Stanford CS229

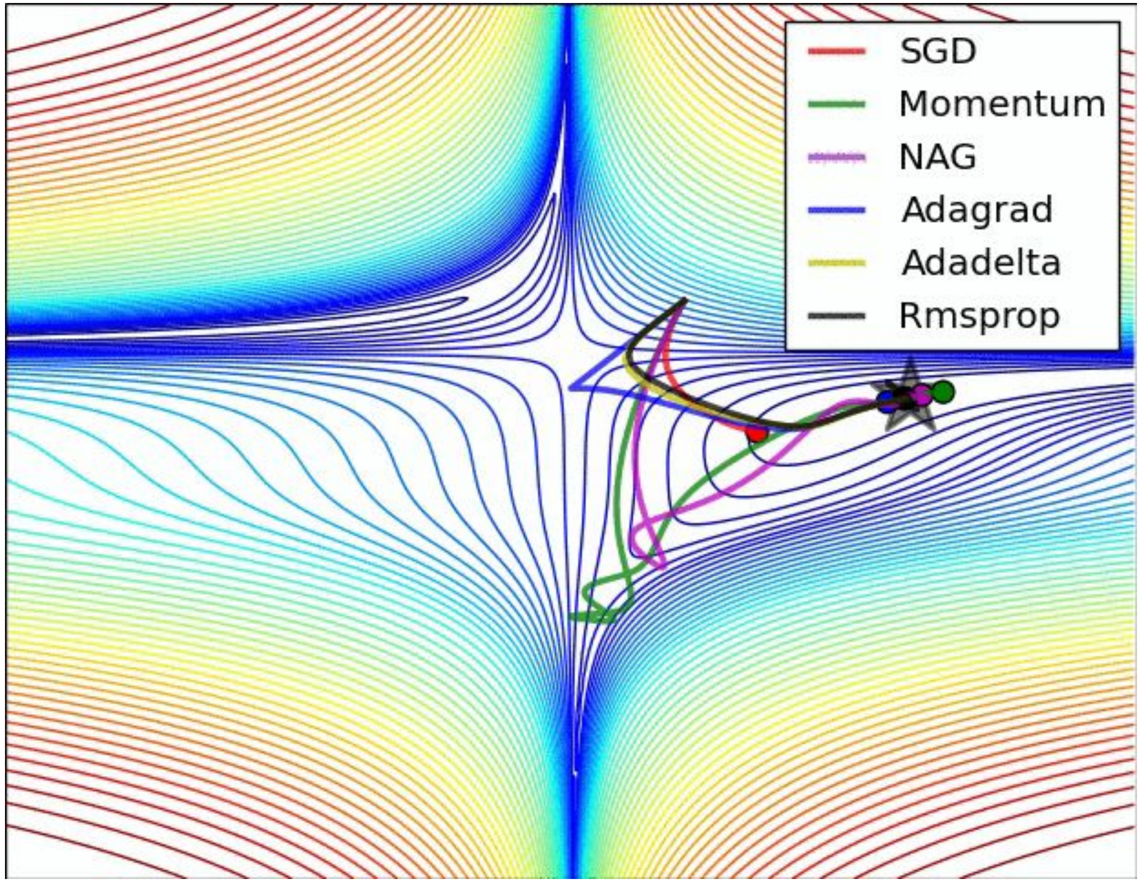
Problems faced with optimizing gradient descent



Different gradient descent optimization algorithms

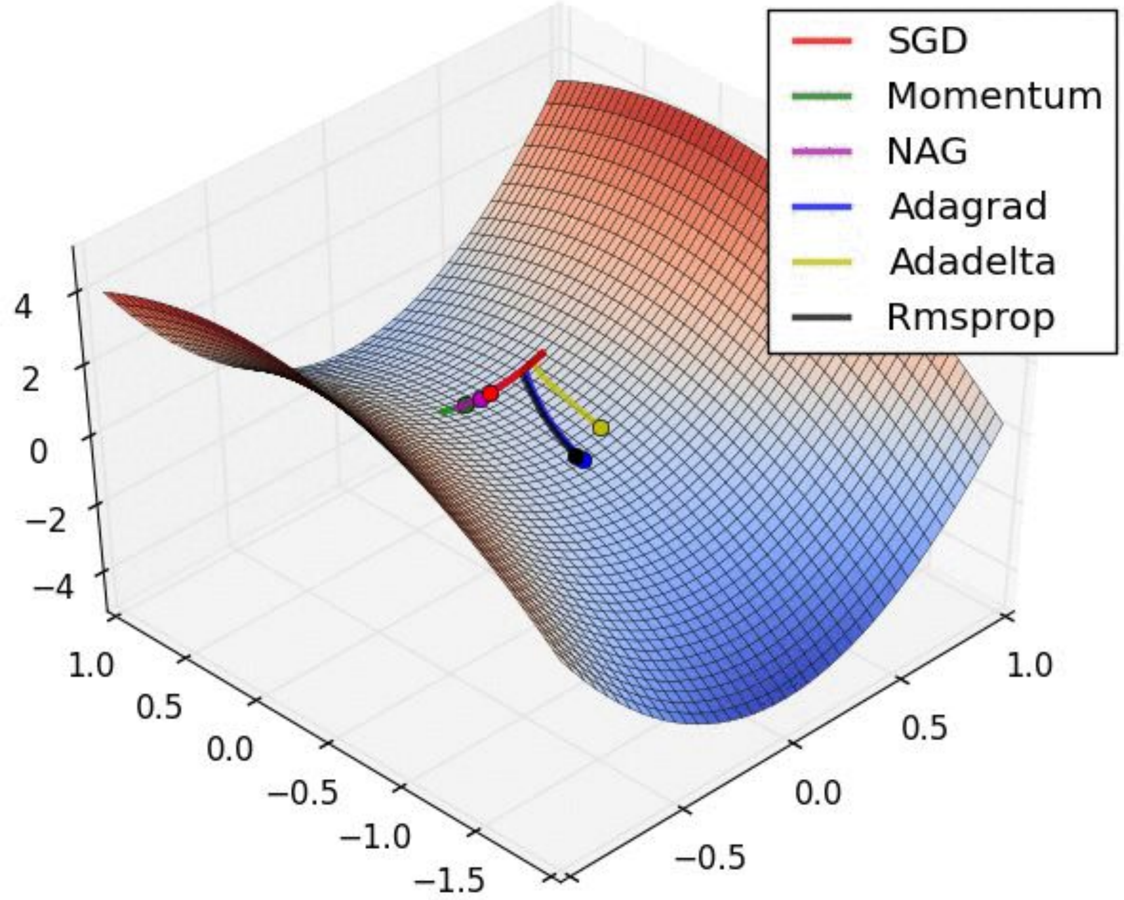
- **Momentum** – adds a fraction of the update vector of the previous time step to the current update vector
- **Nesterov accelerated gradient** – using the momentum term to approximate the next position of the parameters, giving prescience to where the parameters are going to be
- **Adaptive learning rate methods** – eg. Adagrad (adapts the learning rate to the parameters, performing larger updates for infrequent parameters and smaller updates for frequent parameters), Adadelta, RMSprop and Adam

Optimizing gradient descent



Source: Alec Radford

Optimizing gradient descent



Source: Alec Radford