ML1 – introduction

Machine Learning – Tools and applications for policy – Lecture 2

Iman van Lelyveld – Michiel Nijhuis

DNB Data Science Hub

ML1 – introduction

1. What is ML? Can we see OLS as a ML problem?
2. What is ML applied to?
3. The outlines of the ML approach
   - supervised vs. unsupervised learning
   - (hyper)parameters and models
   - gradient descent and grid search
   - pre-processing features

Outline

ML1 – introduction

- Defining ML/AI
- The ML approach
- OLS with Gradient Descent
- Scaling, normalization and standardization
- The workhorse: the logit activation function
- Ensuring robustness and measure performance

Knowledge clips

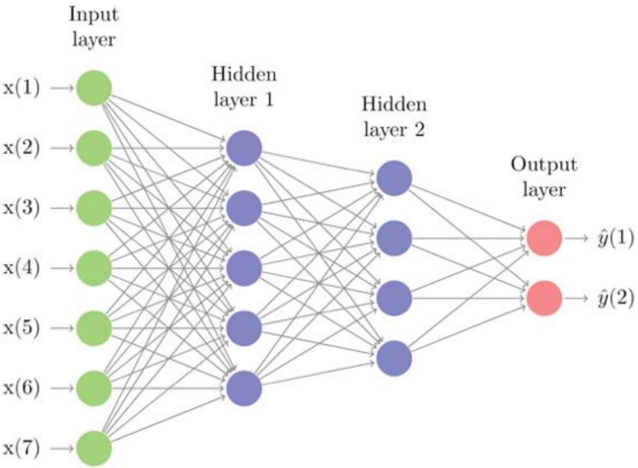
- How does optimization work? Part 1 (Brandon Rohrer) (link)
- How does optimization work? Part 2 (Brandon Rohrer) (link)
- How does optimization work? Part 3 (Brandon Rohrer) (link)
- How does optimization work? Part 4 (Brandon Rohrer) (link)

ML – definitions

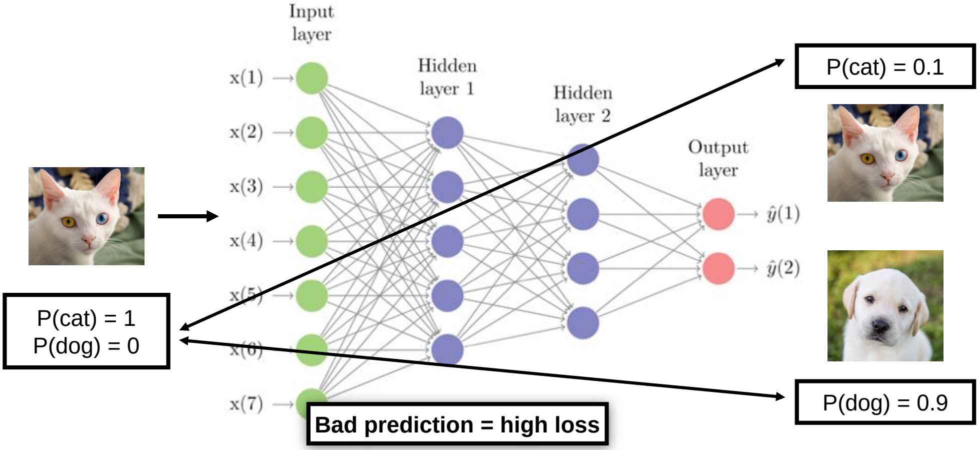
The field of study that gives computers the ability to learn without being explicitly programmed. (Samuel (1967))

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (Mitchell (1997))

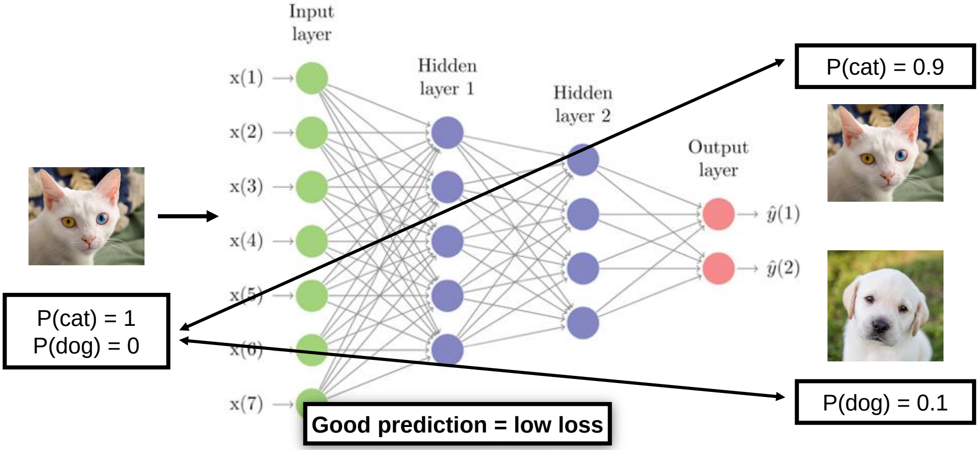
Neural network example of the ML approach



Neural network example of the ML approach



Neural network example of the ML approach



Outline

ML1 – introduction
- Defining ML/AI
- The ML approach
- OLS with Gradient Descent
- Scaling, normalization and standardization
- The workhorse: the logit activation function
- Ensuring robustness and measure performance

Every ML analysis consists of:

1. a problem
2. a data source
3. a model
   - e.g. logit, artificial neural networks
   - cost function: e.g. MSE
   - activation function
   - regularization scheme
4. an optimization algorithm
   - out of scope
   - Out of the box `Python` or `R` optimizers can get stuck in a local minimum
   - Initialization
5. validation & testing

ML – more formally

- ML is an optimization problem, whose solution determines a set of model parameters $F(X, Y, \beta; \lambda)$
  - Y is the target/outcome variable that you want to predict with input data X
  - λ are model parameters depending on the input and the model class
  - Optimizing the objective/cost function $F(\cdot, \beta)$, wrt the parameters λ(training)

ML – more formally

- ML is an optimization problem, whose solution determines a set of model parameters $F(X, Y, \beta; \lambda)$
  - Y is the target/outcome variable that you want to predict with input data X
  - λ are model parameters depending on the input and the model class
  - Optimizing the objective/cost function $F(\cdot, \beta)$, wrt the parameters λ(training)
- Hyper-parameter(s) λ are used to improve model test performance. Performance is determined via validation (calibration) and a popular choice is the smoothing of model output (regularization).

ML – more formally

- ML is an optimization problem, whose solution determines a set of model parameters $F(X, Y, \beta; \lambda)$
  - Y is the target/outcome variable that you want to predict with input data X
  - λ are model parameters depending on the input and the model class
  - Optimizing the objective/cost function $F(\cdot, \beta)$, wrt the parameters λ(training)
- Hyper-parameter(s) λ are used to improve model test performance. Performance is determined via validation (calibration) and a popular choice is the smoothing of model output (regularization).
- Although terminology in machine learning and econometrics varies, it often refers to the same concepts.
  - dependent variable, left-hand-side (LHS) $\equiv$ Output, target response
  - independent variables, covariates, right-hand-side (RHS) $\equiv$ Features, inputs

ML – more formally

- ML is an optimization problem, whose solution determines a set of model parameters $F(X, Y, \beta; \lambda)$
  - Y is the target/outcome variable that you want to predict with input data X
  - λ are model parameters depending on the input and the model class
  - Optimizing the objective/cost function $F(\cdot, \beta)$, wrt the parameters λ(training)
- Hyper-parameter(s) λ are used to improve model test performance. Performance is determined via validation (calibration) and a popular choice is the smoothing of model output (regularization).
- Although terminology in machine learning and econometrics varies, it often refers to the same concepts.
  - dependent variable, left-hand-side (LHS) $\equiv$ Output, target response
  - independent variables, covariates, right-hand-side (RHS) $\equiv$ Features, inputs
- #features does not have to equal #X $\rightarrow$ higher-order features

ML versus Econometric mindset

We may define a cause to be an object, followed by another, and where all the objects similar to the first are followed by objects similar to the second. Or in other words where, if the first object had not been, the second never had existed (Hume (1748))

Social scientists know that large amounts of data will not overcome the selection problems that make causal inference so difficult (Grimmer, Justin (2015))

ML versus Econometric mindset

We may define a cause to be an object, followed by another, and where all the objects similar to the first are followed by objects similar to the second. Or in other words where, if the first object had not been, the second never had existed (Hume (1748))

Social scientists know that large amounts of data will not overcome the selection problems that make causal inference so difficult (Grimmer, Justin (2015))

- Econometrics:
  - Classical conditional: if X occurred, then Y occurred
  - Counterfactual: if X had not occurred, then Y would not have occurred
  - Model and causality imply assumptions about the error term
- Machine Learning:
  - Prediction/classification
  - "If it works, it works"

Common choices for components of the ML design

Feature	Role	Most common	Other examples in practice
Cost function	Calculate penalty/error in prediction versus true output	Mean squared error (regression), Binary cross-entropy (classification)	Mean absolute error, Categorical cross entropy, Kullback-Leibler divergence, Cosine proximity, Hinge/Squared-Hinge, log-cosh

Based on Kolanovic and Krishnamachari (2017)

Common choices for components of the ML design

Feature	Role	Most common	Other examples in practice
Cost function	Calculate penalty/error in prediction versus true output	Mean squared error (regression), Binary cross-entropy (classification)	Mean absolute error, Categorical cross entropy, Kullback-Leibler divergence, Cosine proximity, Hinge/Squared-Hinge, log-cosh
Activation Function	Achieve non-linear effect following each neuron (after the weighted linear combination)	ReLU (intermediate layers), Linear (final layer in regression), Sigmoid (final layer in classification)	Softmax/Softplus/Softsign, Leaky/Parametrized ReLU, tanh, Hard Sigmoid

Based on Kolanovic and Krishnamachari (2017)

Common choices for components of the ML design

Feature	Role	Most common	Other examples in practice
Cost function	Calculate penalty/error in prediction versus true output	Mean squared error (regression), Binary cross-entropy (classification)	Mean absolute error, Categorical cross entropy, Kullback-Leibler divergence, Cosine proximity, Hinge/Squared-Hinge, log-cosh
Activation Function	Achieve non-linear effect following each neuron (after the weighted linear combination)	ReLU (intermediate layers), Linear (final layer in regression), Sigmoid (final layer in classification)	Softmax/Softplus/Softsign, Leaky/Parametrized ReLU, tanh, Hard Sigmoid
Regularization Scheme	Penalize large weights to avoid overfitting	Dropout	L1/L2 regularization for kernel, bias and activation

Based on Kolanovic and Krishnamachari (2017)

Common choices for components of the ML design

Feature	Role	Most common	Other examples in practice
Cost function	Calculate penalty/error in prediction versus true output	Mean squared error (regression), Binary cross-entropy (classification)	Mean absolute error, Categorical cross entropy, Kullback-Leibler divergence, Cosine proximity, Hinge/Squared-Hinge, log-cosh
Activation Function	Achieve non-linear effect following each neuron (after the weighted linear combination)	ReLU (intermediate layers), Linear (final layer in regression), Sigmoid (final layer in classification)	Softmax/Softplus/Softsign, Leaky/Parametrized ReLU, tanh, Hard Sigmoid
Regularization Scheme	Penalize large weights to avoid overfitting	Dropout	L1/L2 regularization for kernel, bias and activation
Optimizer	Calibrate network weights based on error	Stochastic Gradient Descent (SGD)	RMSprop54, Adagrad, Adadelta, Adam, Adamax, Nesterov-Adam

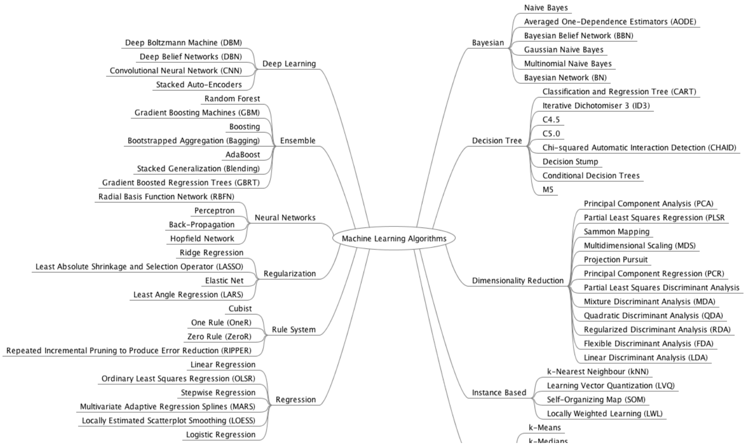
Based on Kolanovic and Krishnamachari (2017)

Common choices for components of the ML design

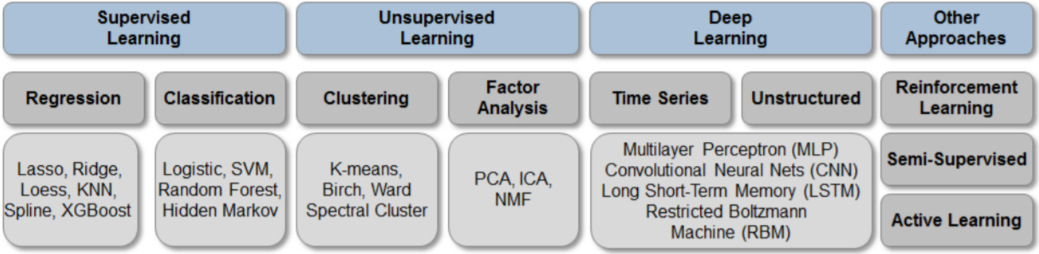
Feature	Role	Most common	Other examples in practice
Cost function	Calculate penalty/error in prediction versus true output	Mean squared error (regression), Binary cross-entropy (classification)	Mean absolute error, Categorical cross entropy, Kullback-Leibler divergence, Cosine proximity, Hinge/Squared-Hinge, log-cosh
Activation Function	Achieve non-linear effect following each neuron (after the weighted linear combination)	ReLU (intermediate layers), Linear (final layer in regression), Sigmoid (final layer in classification)	Softmax/Softplus/Softsign, Leaky/Parametrized ReLU, tanh, Hard Sigmoid
Regularization Scheme	Penalize large weights to avoid overfitting	Dropout	L1/L2 regularization for kernel, bias and activation
Optimizer	Calibrate network weights based on error	Stochastic Gradient Descent (SGD)	RMSprop54, Adagrad, Adadelta, Adam, Adamax, Nestorov-Adam
Initialization Scheme	Initialize network weights	Xavier (including Glorot-Normal and Glorot-Uniform)	0s/1s/Constant, Random Normal/Uniform, Variance Scaling, Orthogonal, Le Cun-Uniform, He-Normal/Uniform

Based on Kolanovic and Krishnamachari (2017)

Taxonomy of the methods



Taxonomy of the methods



Which model to use?

model class	S/US	reg/class	parametric	data size	norm	suited for	advantages	disadvantages
OLS	S	reg	yes	small - large	no	simple relations, hypothesis testing	interpretability, computability	inflexibility
Logit	S	class	yes	small - large	no	simple relations, hypothesis testing	interpretability, computability	inflexibility
naïve Bayes	S	class	no	small - large	no	simple benchmark	computability	independence assumption
k-NN	S	reg/class	no	small - medium	yes	clustered data, multiple regimes	flexibility	interpretability, COD
tree model	S	reg/class	no	small - large	no	complex relations, multiple regimes	flexibility, interpretability, computability	greedy, over-fitting
random forest	S	reg/class	no	small - medium	no	complex relations, multiple regimes	flexibility	computability, interpretability
artificial neural network (ANN)	S	reg/class	semi	mid - large	yes	complex relations, multiple scales	flexibility	computability, over-fitting, data hungry, interpretability
support vector machine (SVM)	S	reg/class	no	small - medium	yes	complex relations	flexibility, computability	over-fitting, interpretability
k-means	US	- -	no	small - large	yes	feature extraction, stylised facts, structure	interpretability	interpretability, COD
hierarchical clustering analysis (HCA)	US	- -	no	small - large	yes	feature extraction, stylised facts, structure	interpretability	interpretability, COD

Source: Chakraborty and A. Joseph (2017)

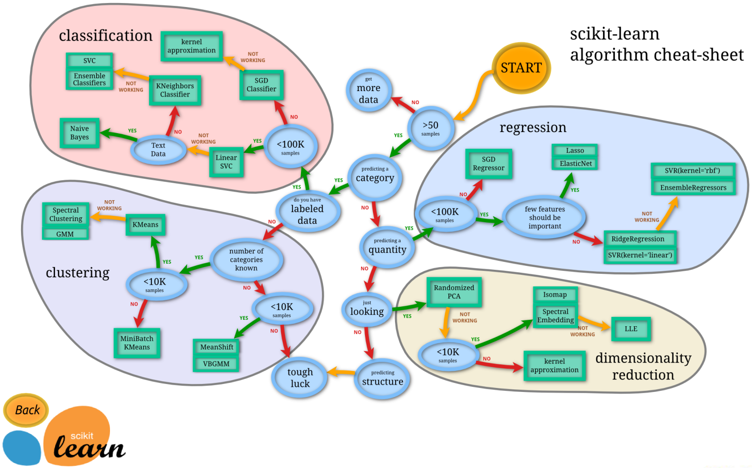
Which model to use?

Figure 41: Typical tasks and frequently used Machine Learning methods

Question	Data Analysis Technique
Given set of inputs, predict asset price direction	Support Vector Classifier, Logistic Regression, Lasso Regression, etc.
How will a sharp move in one asset affect other assets?	Impulse Response Function, Granger Causality
Is an asset diverging from other related assets?	One-vs-rest classification
Which assets move together?	Affinity Propagation, Manifold Embedding
What factors are driving asset price?	Principal Component Analysis, Independent Component Analysis
Is the asset move excessive, and will it revert?	
What is the current market regime?	Soft-max classification, Hidden Markov Model
What is the probability of an event?	Decision Tree, Random Forest
What are the most common signs of market stress?	K-means clustering
Find signals in noisy data	Low-pass filters, SVM
Predict volatility based on a large number of input variables	Restricted Boltzmann Machine, SVM
What is the sentiment of an article / text?	Bag of words
What is the topic of an article/text?	Term/InverseDocument Frequency
Counting objects in an image (satellite, drone, etc)	Convolutional Neural Nets
What should be optimal execution speed?	Reinforcement Learning using Partially Observed Markov Decision Process

Source: J.P.Morgan Macro QDS

Which model to use?



Source: SK Learn website

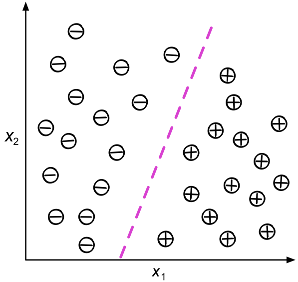
Some order: Supervized learning

- classical $Y = f(X)$
- *labeled*: x_i is matched with y_i. Alternatively, it is known what class y_i belongs to (eg employed/unemployed)

1. Classification problems
   - is it a cat or a dog?
   - output Y consists of a discrete set of outcomes which cannot be ordered. That is, each element of Y represents a class label
   - economics: employment status of individuals (Y: employed or unemployed) from their communication or consumption habits ($X_{i..J}$) would be a typical classification problem. A learning method would return one of the two states/labels for each observation x_i
2. Regression problems
   - match and return a continuous output variable

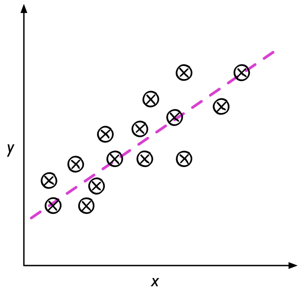
Supervised - Classification

- Predict categorical class labels based on past observations
- Class labels are discrete unordered values which cannot be ordered.
- That is, each element of $\mathcal{Y}$ represents a class label and output $\mathcal{Y}$ consists of a discrete set of outcomes
- Examples
  - Binary: Email spam classification example or unemployment status
  - Multi-class: Handwritten digit classification example



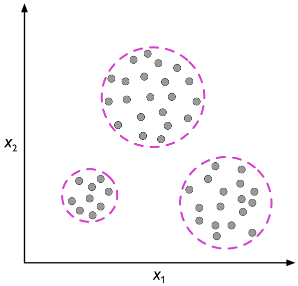
Supervised - Regression

- Also a kind of supervised learning
- Prediction of continuous outcomes
- Predicting semester grades scores for students



Unsupervised learning

- Finding structure in the data, e.g. clustering
- Objects within a cluster are "similar"
- Dealing with unlabeled data
- Can be used first to add labels to the observations or extract new features (e.g. cluster affiliation)
- Examples:
  - Cluster analysis
  - Latent Dirichlet Allocation (LDA)
  - Princeton Wiki
  - Wikipedia LDA entry
  - Sara Palin topics



Outline

ML1 – introduction
- Defining ML/AI
- The ML approach
- OLS with Gradient Descent
- Scaling, normalization and standardization
- The workhorse: the logit activation function
- Ensuring robustness and measure performance

Understanding ML using *Gradient Descent* for OLS

- The defining characteristic: taking the target variable Y as an input to the cost function
- Given a hypothesis $h(X)$ representing the model, a commonly-used objective/cost function is the mean squared error (MSE). Problem takes the general form:

$$ERR(X, Y, \beta) \overset{\text{MSE}}{=} \frac{1}{m}\sum_{i=1}^{m} e_i^2 \equiv \frac{1}{m}\sum_{i=1}^{m}\Big(h(x_i, \beta) - y_i\Big)^2 \xrightarrow[\text{algorithm}]{\text{optimisation}} \beta\,,$$

- The more data we have, the more complicated $h(\dot{})$ can be: logit can be exchanged for a deep neural network

See "How does optimization work" in Knowledge clips.

Gradient descent

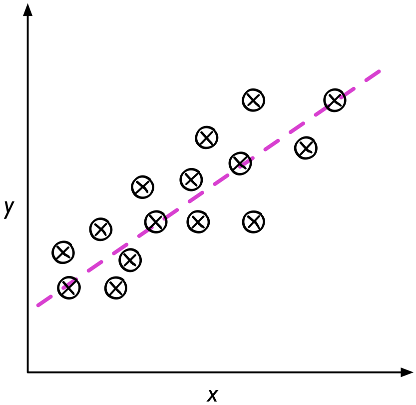
- Linear regression:

$$h(X) = X \cdot \beta \quad \rightarrow \quad ERR(X, Y, \beta) = \frac{1}{m} \sum_{i=1}^{m} \left(x_i \cdot \beta - y_i \right)^2$$

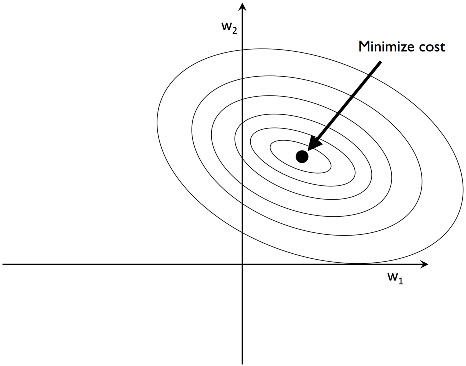
- With the Gauss-Markov theorem, the best linear unbiased estimator (BLUE) has the closed-form solution: $\beta^* = (X^T X)^{-1} X^T Y$

- BLUE assumptions might be violated $\rightarrow$ optimisation
- Algorithms look for a trade-off between computational cost and accuracy

Regression



Animated version



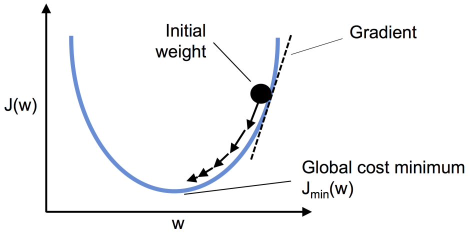
Updating weights

- We apply this procedure a number of times, and in each iteration we update the weights s.t.

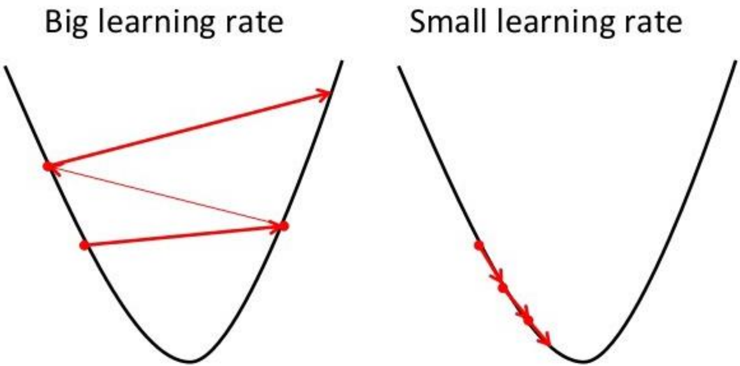
$$w_i := w_i - \alpha \frac{\delta J(w)}{\delta_i}$$

where α is a value in (0;1] called the learning rate

- The algorithm is called the (batch) gradient descent (GD)
- The loss (and therefore the gradient) is computed over all observations

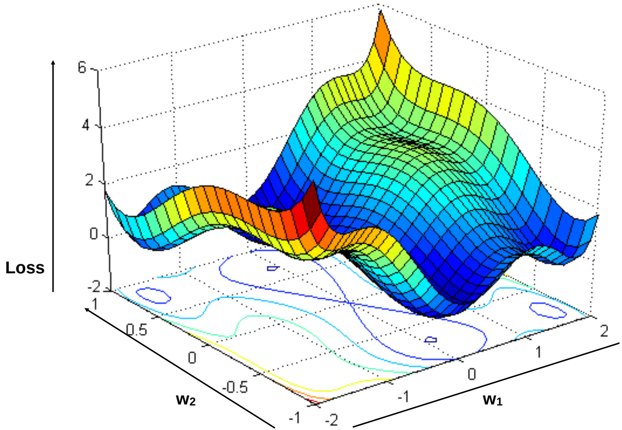


Learning rate



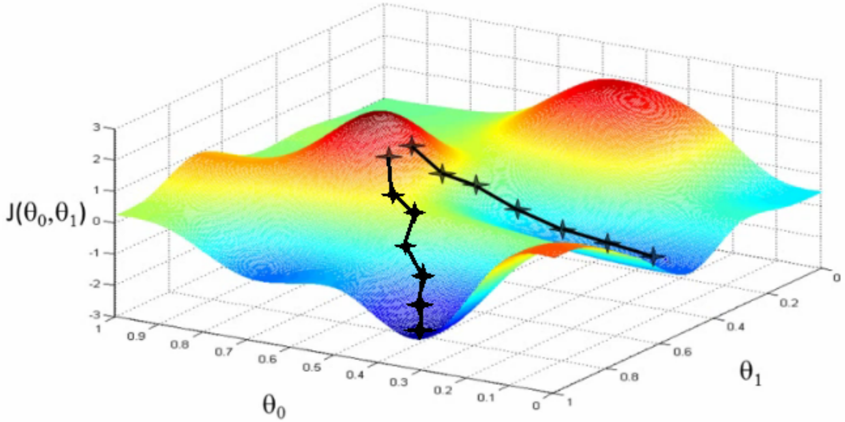
Source: What are Hyperparameters

A more realistic loss function



Source: An introduction to loss functions

Gradient Descent: remember the intuition



Source: analyticsvidhya.com

Stochastic Gradient Descent (SGD)

- Large dataset with millions of data points ("big data") then batch gradient descent costly
- Remember that 'batch' actually means all the data so we need to compute the error for the entire dataset ...
- ... to take one step towards the global minimum!

$$\Delta \mathbf{w} = \eta \sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) \mathbf{x}^{(i)}.$$

- SGD updates the weights incrementally for each training sample

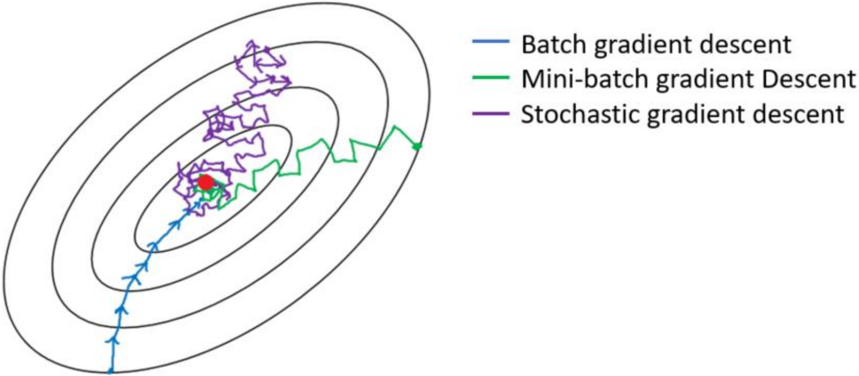
$$\Delta \mathbf{w} = \eta \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) \mathbf{x}^{(i)}.$$

SGD details

- Approximation of gradient descent
- Faster convergence because of frequent weight updates
- Important to present data in random order
- Learning rate often gradually decreased (adaptive learning rate)
- Can be used for online learning

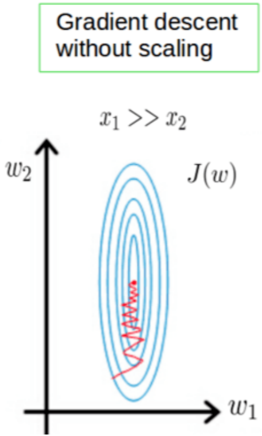
- Middle ground between SGD and batch GD is known as mini-batch learning
  - E.g. 50 examples at a time
  - Can use vector/matrix operations rather than loops as in SGD
  - Vectorized operations highly efficient
- SGD is mini-batch with $n = 1$

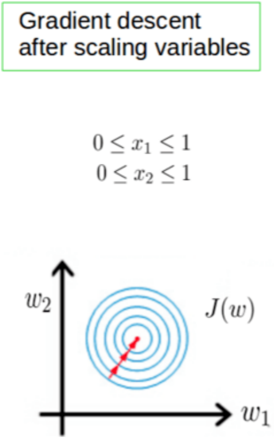
Batch size



Source: Andrew Ng – Coursera Course

Why scale features: the intuition





DataScience Hub

Feature scaling

- So Input preprocessing is important but may crucially affect a learners' performance
- Imagine we have two features
  - Feature x_1: $1 < x_1 < 10$
  - Feature x_2: $1 < x_2 < 100000$
- Algorithm will likely focus on optimizing w_2 as this will produce the largest changes in e.g. error/loss function. K-nearest neighbor (KNN, Euclidean distance) and Artificial Neural Networks (ANN) will be dominated by x_2. (Both KNN and ANN discussed later)
- Two common approaches
  - Normalization
  - Standardization
    - with μ and σ
    - with σ
- See an excellent article by Jeff Hale with JPNB. Note differences in definition.

Normalization

Normalization refers to the rescaling of the features to a range of [0, 1]. To normalize the data, we apply the min-max scaling to each feature column, where the new value $x_{norm}^{(i)}$ of a sample $x^{(i)}$ is calculated as follows:

$$x_{norm}^{(i)} = \frac{x^{(i)} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}$$

Here, $x^{(i)}$ is a particular sample/observation, x_{min} is the smallest value in a feature column, and x_{max} the largest value, respectively.

Standardization

- Standardization (i.e., *z*-scores) centers the columns at $mean = 0$ and $std = 1$

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

or similarly rescale with just σ

Standardization

- Standardization (i.e., *z*-scores) centers the columns at $mean = 0$ and $std = 1$

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

  or similarly rescale with just σ

- While normalization gives us values in a bounded interval, standardization can be more practical
  - Many ML algorithms initialize the weights to zero
  - Feature columns take the form of a normal distribution
  - This makes it easier to learn the weights
- Standardization encodes useful info about outliers
- Normalization – in contrast – scales the data to a fixed range

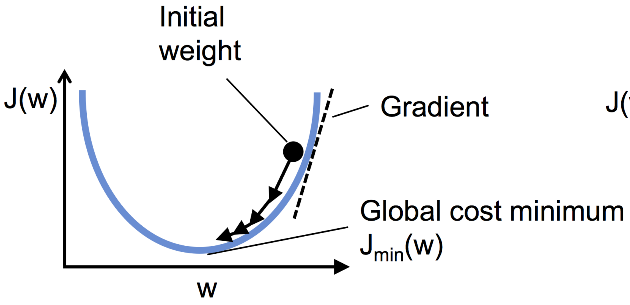
Standardization

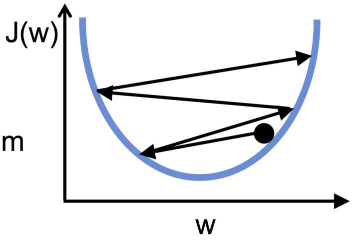
- Standardization (i.e., *z*-scores) centers the columns at $mean = 0$ and $std = 1$

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

  or similarly rescale with just σ
- While normalization gives us values in a bounded interval, standardization can be more practical
  - Many ML algorithms initialize the weights to zero
  - Feature columns take the form of a normal distribution
  - This makes it easier to learn the weights
- Standardization encodes useful info about outliers
- Normalization – in contrast – scales the data to a fixed range
- Binning can also address non-linearity/heterogeneity (Finlay (2014))
- Other ways to achieve approximate standard normality: log or a Box-C… transformation (cf A. C. Joseph et al. (2014))

IMPORTANT Lessons





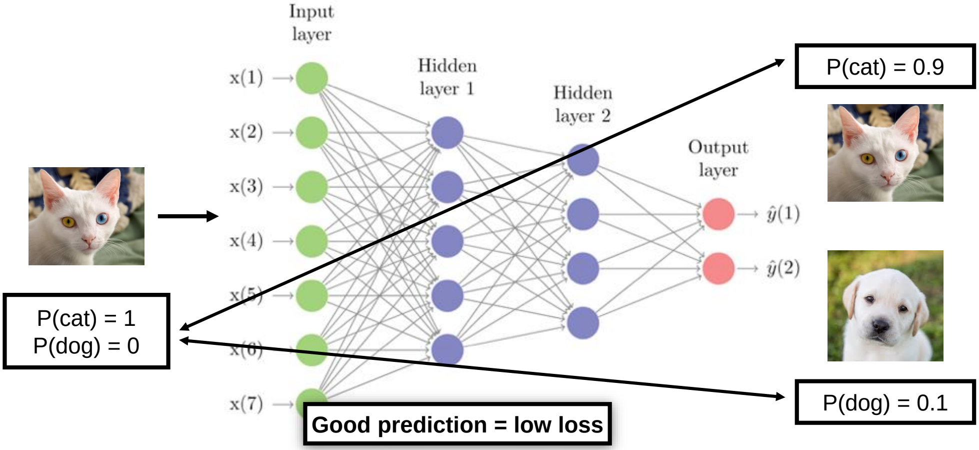
- Learning rate too high: error becomes larger (overshoots global min)
- Learning rate too low: takes many epochs to converge
- Feature normalization

Outline

ML1 – introduction

- Defining ML/AI
- The ML approach
- OLS with Gradient Descent
- Scaling, normalization and standardization
- The workhorse: the logit activation function
- Ensuring robustness and measure performance

Reminder: Neural network example of the ML approach



Logistic regression model

- This is a "go to" model for classification
- Designed for binary classification but can be extended to multi-class
- Odds ratio

$$\frac{p}{(1-p)}$$

Where p is the probability of the positive class (class label $y = 1$). E.g. the probability that a patient has a certain disease.

- Logit function

$$logit(p) = \log \frac{p}{(1-p)}$$

Modeling logit function

- We model the logit function as a linear combination of features (dot product of feature values and weights)

$$logit(p(y=1|\mathbf{x})) = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i = \mathbf{w}^T\mathbf{x}.$$

  Where $p(y=1|\mathbf{x})$ s the conditional probability that a particular sample belongs to class 1 given its features $\mathbf{x}$

- This is equivalent to expressing p as

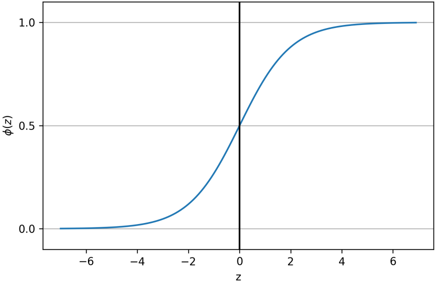
$$p(y=1|\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

Logistic Sigmoid

- Logistic function (aka sigmoid function)

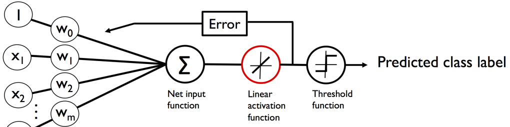
$$\phi(z) = \frac{1}{1 + e^{-z}}.$$

- S-shaped curve

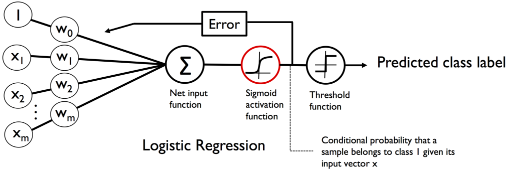


Relationship with Adaptive Linear Neuron (Adaline)

- In Adaline, we used the “identity” function as the activation function
- In logistic regression, we use instead use the “sigmoid” function



Adaptive Linear Neuron (Adaline)



Logistic Regression

Many activation functions around

Sigmoid logistic $y = \frac{1}{1+e^{-z}}$

Tanh hyperbolic tangent $y = tanh(z)$

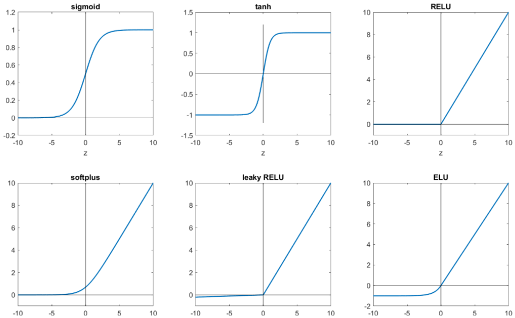
ReLU Rectified Linear Unit $y = max(z, 0)$

Softplus $y = log(1 + e^z)$

Leaky Regularization $y = max(z, \alpha z), 0 < \alpha < 1$

α typically a small number e.g. 0.01

Elu $y = \begin{cases} z & z > 0 \\ \alpha(e^z - 1) & z \leq 0 \end{cases}$



Many more functions can be found here

Experiment with different functions in this Colab notebook

Probability distribution over classes for a Logit model

- Output of the sigmoid often interpreted as probability
- E.g. $P(y = 1|\mathbf{x}; \mathbf{w}) = 0.8$
- Probability can be converted to a binary outcome (quantizer)

$$\hat{y} = \begin{cases} 1 & \text{if } \phi(z) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- Which is equivalent to the following

$$\hat{y} = \begin{cases} 1 & \text{if } z \geq 0.0 \\ 0 & \text{otherwise} \end{cases}$$

- For many applications (e.g. weather forecasting, default), we want the probability

Learning the weights

- Previously we minimized the sum-squared-error cost function

$$J(\mathbf{w}) = \frac{1}{2}\sum_i \left(\phi(z^{(i)}) - y^{(i)}\right)^2$$

- Now we need to derive the cost function for logistic regression
- Define the likelihood L

$$L(\mathbf{w}) = P(\mathbf{y}|\mathbf{x};\mathbf{w}) = \prod_{i=1}^{n} P\left(y^{(i)}|x^{(i)};\mathbf{w}\right)$$

$$L(\mathbf{w}) = \prod_{i=1}^{n} \left(\phi(z^{(i)})\right)^{y^{(i)}} \left(1 - \phi(z^{(i)})\right)^{1-y^{(i)}}$$

Log-likelihood function

- Maximize the likelihood function

$$L(\mathbf{w}) = P(\mathbf{y}|\mathbf{x};\mathbf{w})$$

$$L(\mathbf{w}) = \prod_{i=1}^{n} \left(\phi\left(z^{(i)}\right)\right)^{y^{(i)}} \left(1 - \phi\left(z^{(i)}\right)\right)^{1-y^{(i)}}$$

- In practice easier to deal with the natural log of this equation

$$l(\mathbf{w}) = \log L(\mathbf{w})$$

$$l(\mathbf{w}) = \sum_{i=1}^{n} \left[y^{(i)} \log\left(\phi\left(z^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - \phi\left(z^{i()}\right)\right) \right]$$

- Easier to take derivative + fewer numerical underflow issues

Cost function

- Rewrite likelihood as a cost function

$$J(\mathbf{w}) = \sum_{i=1}^{n} \left[-y^{(i)} \log\left(\phi(z^{(i)})\right) - \left(1 - y^{(i)}\right) \log\left(1 - \phi(z^{i()})\right) \right]$$

- which can now be minimized using gradient descent
- Derivation in the slide at the end

▸ iPython notebook on github

DataScience Hub

Ahead: How to ensure robustness and measure performance?

- Great that we have methods to come to a prediction or a classification but how do we know if this makes any sense?
- In "ML2 – the basics" we will discuss in detail:
  -
  - Bias-Variance
  - Overfitting
  - Test-Train
  - Confusion matrix
  - ...

		Predicted class	
		P	N
Actual class	P	True positives (TP)	False negatives (FN)
	N	False positives (FP)	True negatives (TN)

Ahead: How to ensure robustness and measure performance?

- Great that we have methods to come to a prediction or a classification but how do we know if this makes any sense?
- In "ML2 – the basics" we will discuss in detail:
  -
  - Bias-Variance
  - Overfitting
  - Test-Train
  - Confusion matrix
  - ...

False Negatives!

Ahead: How to ensure robustness and measure performance?

- Great that we have methods to come to a prediction or a classification but how do we know if this makes any sense?
- In "ML2 – the basics" we will discuss in detail:
  -
  - Bias-Variance
  - Overfitting
  - Test-Train
  - Confusion matrix
  - ...

False Positives!

Summary

In this lecture we covered:

1. a few examples of how ML is applied in other fields
2. how an Ordinary Least Squares (OLS) model can be seen as a ML model
3. how updating weights leads to learning
4. Discussed the relevance of normalizing or standardizing
5. Worked our way up to the workhorse of classification – the logit model

References I

- Chakraborty, C., & Joseph, A. (2017). Machine learning at central banks. Bank of England Working Paper, 674. Retrieved December 7, 2018, from www.bankofengland.co.uk/research/Pages/workingpapers/default.aspx
- Finlay, S. (2014). Predictive Analytics, Data Mining and Big Data. Palgrave MacMillan.
- Grimmer, Justin. (2015). We Are All Social Scientist Now: How Big Data, Machine Learning, and Causal Inference Work Together. Political Science, 80–83.
- Hume, D. (1748). An Enquiry Concerning Human Understanding.
- Joseph, A. C., Joseph, S. E., & Chen, G. (2014). Cross-border portfolio investment networks and indicators for financial crises.. Scientific reports, 4, 3991.
- Kolanovic, M., & Krishnamachari, R. T. (2017). Big Data and AI Strategies - Machine Learning and Alternative Data Approach to Investing Quantitative and Derivatives Strategy. J.P. Morgan report.
- Mitchell, T. (1997). Machine Learning. McGraw Hill.

References II

Samuel, A. L. (1967). Some Studies in Machine Learning Using the Game of Checkers - Recent Progress. IBM Journal of Research and Development, 11(6), 601–617.