MACHINE LEARNING IN SYSTEMS BIOLOGY I

Jonas Pleyer

29.06.2022

Freiburg Center for Data Analysis and Modeling (FDM)

INTRODUCTION

WHAT IS MACHINE LEARNING?

'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'
- Tom M. Mitchell [Mit97]

HISTORY OF MACHINE LEARNING

1943 First publication of neural network [MP43]

History of machine learning

1943 First publication of neural network [MP43]

1956 Dartmouth Summer Research Project (Birthplace of modern Machine Learning)

History of machine learning

1943 First publication of neural network [MP43]

1956 Dartmouth Summer Research Project (Birthplace of modern Machine Learning)

1965 Nilson Machine Learning for pattern classification [Nil65]

HISTORY OF MACHINE LEARNING

1943 First publication of neural network [MP43]

1956 Dartmouth Summer Research Project (Birthplace of modern Machine Learning)

1965 Nilson Machine Learning for pattern classification [Nil65]

1966 Following years: Many setbacks in Artificial Intelligence called 'AI-Winters'

HISTORY OF MACHINE LEARNING

1943 First publication of neural network [MP43]

1956 Dartmouth Summer Research Project (Birthplace of modern Machine Learning)

1965 Nilson Machine Learning for pattern classification [Nil65]

1966 Following years: Many setbacks in Artificial Intelligence called 'AI-Winters'

1995 Support Vector Machines are first introduced

HISTORY OF MACHINE LEARNING

1943 First publication of neural network [MP43]

1956 Dartmouth Summer Research Project (Birthplace of modern Machine Learning)

1965 Nilson Machine Learning for pattern classification [Nil65]

1966 Following years: Many setbacks in Artificial Intelligence called 'AI-Winters'

1995 Support Vector Machines are first introduced

2002 Torch first release (open source library)

HISTORY OF MACHINE LEARNING

1943 First publication of neural network [MP43]

1956 Dartmouth Summer Research Project (Birthplace of modern Machine Learning)

1965 Nilson Machine Learning for pattern classification [Nil65]

1966 Following years: Many setbacks in Artificial Intelligence called 'AI-Winters'

1995 Support Vector Machines are first introduced

2002 Torch first release (open source library)

2006 Geoffrey Hinton coins 'Deep Learning' [HOT06]

HISTORY OF MACHINE LEARNING

1943	First publication of neural network [MP43]
1956	Dartmouth Summer Research Project (Birthplace of modern Machine Learning)
1965	Nilson Machine Learning for pattern classification [Nil65]
1966	Following years: Many setbacks in Artificial Intelligence called 'AI-Winters'
1995	Support Vector Machines are first introduced
2002	Torch first release (open source library)
2006	Geoffrey Hinton coins 'Deep Learning' [HOT06]
>2006	Companies such as Netflix, Facebook, Microsoft, Google fund projects/prizes in and use machine learning/artificial intelligence

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)
3. Pre-process Data (cleanup, drop)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)
3. Pre-process Data (cleanup, drop)
4. Analyze Data (define features, find correlations)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)
3. Pre-process Data (cleanup, drop)
4. Analyze Data (define features, find correlations)
5. Prepare Data (transform, normalize, drop)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)
3. Pre-process Data (cleanup, drop)
4. Analyze Data (define features, find correlations)
5. Prepare Data (transform, normalize, drop)
6. Evaluate Models (train/test, classify/regress)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)
3. Pre-process Data (cleanup, drop)
4. Analyze Data (define features, find correlations)
5. Prepare Data (transform, normalize, drop)
6. Evaluate Models (train/test, classify/regress)
7. Tune Model (cross validation, fine tune parameters)

WORKFLOW

Machine learning techniques follow a similar workflow.

1. Define Problem (scope, feasability)
2. Gather Data (assumptions, constraints)
3. Pre-process Data (cleanup, drop)
4. Analyze Data (define features, find correlations)
5. Prepare Data (transform, normalize, drop)
6. Evaluate Models (train/test, classify/regress)
7. Tune Model (cross validation, fine tune parameters)
8. Apply model to problems, learn more

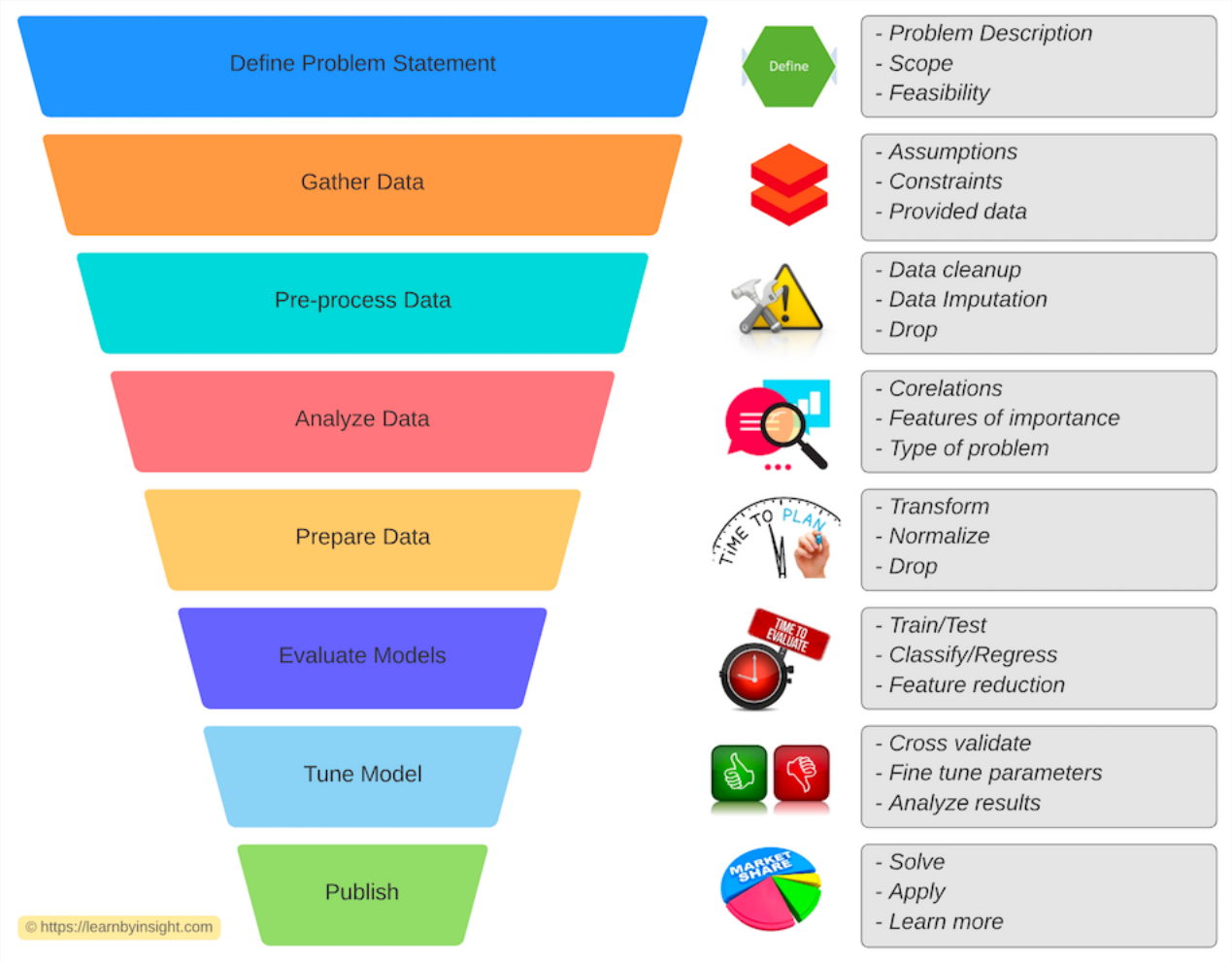

Figure 1: Machine Learning workflow [Mew20]

CONCEPTS

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

Supervised

- Fit model to labelled data (ie. with 'ground truth')

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

- Data does not contain any labels (only inputs)

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

- Data does not contain any labels (only inputs)
- Find structure in data (clustering, grouping)

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

- Data does not contain any labels (only inputs)
- Find structure in data (clustering, grouping)

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

- Data does not contain any labels (only inputs)
- Find structure in data (clustering, grouping)

Semi-supervised

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

- Data does not contain any labels (only inputs)
- Find structure in data (clustering, grouping)

Semi-supervised

- Combine partly labeled data with partly unlabeled data

This section follows [GKMJ21].

UNSUPERVISED AND SUPERVISED LEARNING

Supervised

- Fit model to labelled data (ie. with 'ground truth')
- Data is usually obtained experimentally or assigned by humans
- Previously labelled data can serve as testing set

Unsupvervised

- Data does not contain any labels (only inputs)
- Find structure in data (clustering, grouping)

Semi-supervised

- Combine partly labeled data with partly unlabeled data
- Can have huge performance benefits compared to unsupervised learning

This section follows [GKMJ21].

TERMS

TERMS

- Classification: Assign datapoints discrete categories (eg. cancerous, non-cancerous). Algorithms are called 'classifiers'.

TERMS

- Classification: Assign datapoints discrete categories (eg. cancerous, non-cancerous). Algorithms are called 'classifiers'.

  If discrete categories are mutually exclusive, we call them 'classes', otherwise 'labels'.

TERMS

- Classification: Assign datapoints discrete categories (eg. cancerous, non-cancerous). Algorithms are called 'classifiers'.

  If discrete categories are mutually exclusive, we call them 'classes', otherwise 'labels'.
- Regression: Output continuous values (eg. predict free energy of protein system).

TERMS

- Classification: Assign datapoints discrete categories (eg. cancerous, non-cancerous). Algorithms are called 'classifiers'.

  If discrete categories are mutually exclusive, we call them 'classes', otherwise 'labels'.
- Regression: Output continuous values (eg. predict free energy of protein system).
- Classification problems can also be solved with regression and thresholds/binning.

TERMS

- Classification: Assign datapoints discrete categories (eg. cancerous, non-cancerous). Algorithms are called 'classifiers'.

  If discrete categories are mutually exclusive, we call them 'classes', otherwise 'labels'.
- Regression: Output continuous values (eg. predict free energy of protein system).
- Classification problems can also be solved with regression and thresholds/binning.
- Clustering: Predict groupings of similar datapoints.

TERMS

- Loss or Cost function:
  Measure deviation to ground truth in supervised learning.
  Implemented similarly in unsupervised situations.

TERMS

- Loss or Cost function:
  Measure deviation to ground truth in supervised learning.
  Implemented similarly in unsupervised situations.
- Parameters: Part of the model, will be adjusted by learning process of the model.

TERMS

- Loss or Cost function:
  Measure deviation to ground truth in supervised learning.
  Implemented similarly in unsupervised situations.
- Parameters: Part of the model, will be adjusted by learning process of the model.
- Hyperparameters: Not part of the model but control learning process (eg. learning rate, number of iterations)

TERMS

- Loss or Cost function:
  Measure deviation to ground truth in supervised learning.
  Implemented similarly in unsupervised situations.
- Parameters: Part of the model, will be adjusted by learning process of the model.
- Hyperparameters: Not part of the model but control learning process (eg. learning rate, number of iterations)
- Training: describes process of iterative learning and adjusting the parameters of the model to obtain better performance. Minimize the loss/cost-function.

TERMS

- Loss or Cost function:
  Measure deviation to ground truth in supervised learning.
  Implemented similarly in unsupervised situations.
- Parameters: Part of the model, will be adjusted by learning process of the model.
- Hyperparameters: Not part of the model but control learning process (eg. learning rate, number of iterations)
- Training: describes process of iterative learning and adjusting the parameters of the model to obtain better performance. Minimize the loss/cost-function.
- Validation: Use seperate dataset to test model.

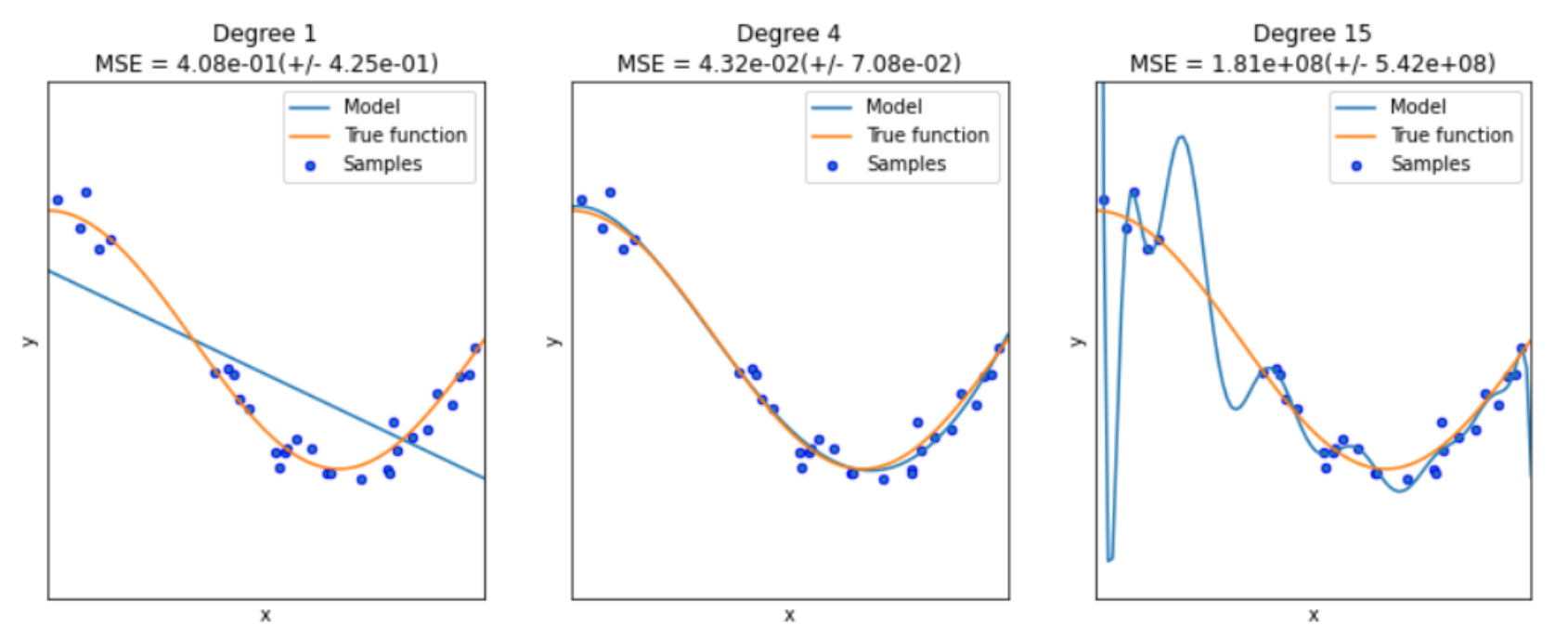

Figure 2: Underfitting, Optimal Fitting and Overfitting [Tri20]

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance
  Different inductive biases typically lead to better performance, but higher constraints on the model.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance
  Different inductive biases typically lead to better performance, but higher constraints on the model.
  Lower bias makes fewer assumptions.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance
  Different inductive biases typically lead to better performance, but higher constraints on the model.
  Lower bias makes fewer assumptions.
- Variance: How much does trained model change in response to training on different dataset.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance
  Different inductive biases typically lead to better performance, but higher constraints on the model.
  Lower bias makes fewer assumptions.
- Variance: How much does trained model change in response to training on different dataset.
- We want low bias and low variance.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance
  Different inductive biases typically lead to better performance, but higher constraints on the model.
  Lower bias makes fewer assumptions.
- Variance: How much does trained model change in response to training on different dataset.
- We want low bias and low variance.
- Low bias and low variance often conflict each other.

INDUCTIVE BIAS AND VARIANCE

- Inductive Bias: Set of assumptions.
  Leads it to favour a particular type of solution over others.
  Often programmed in mathematical model.
  Example: Recurrent Neural Networks anticipate sequential dependencies
- Trade-off between bias and variance
  Different inductive biases typically lead to better performance, but higher constraints on the model.
  Lower bias makes fewer assumptions.
- Variance: How much does trained model change in response to training on different dataset.
- We want low bias and low variance.
- Low bias and low variance often conflict each other.
  $\Rightarrow$ Need to balance between them

MACHINE LEARNING TECHNIQUES

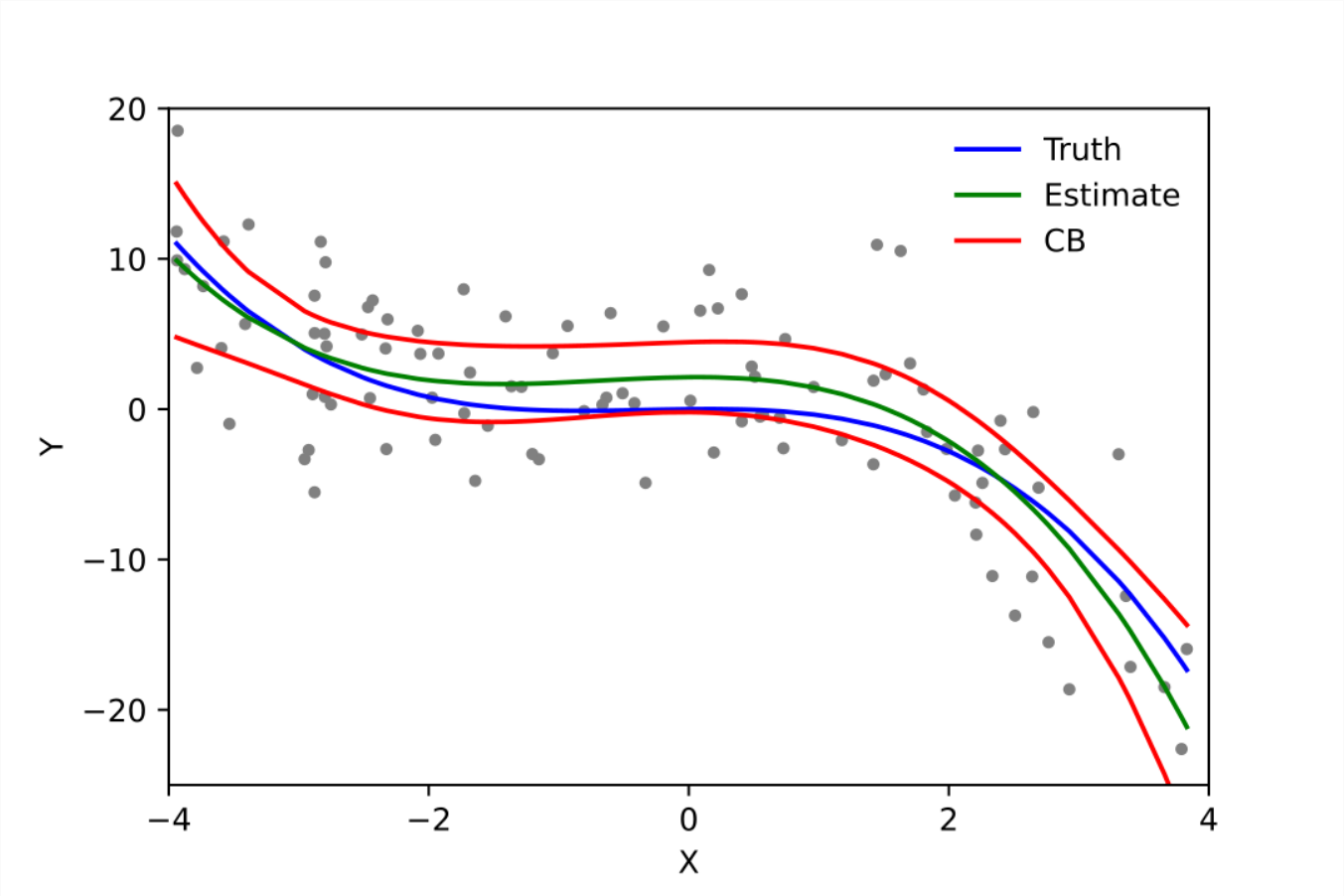

Figure 3: Polynomial Regression [Skb09]

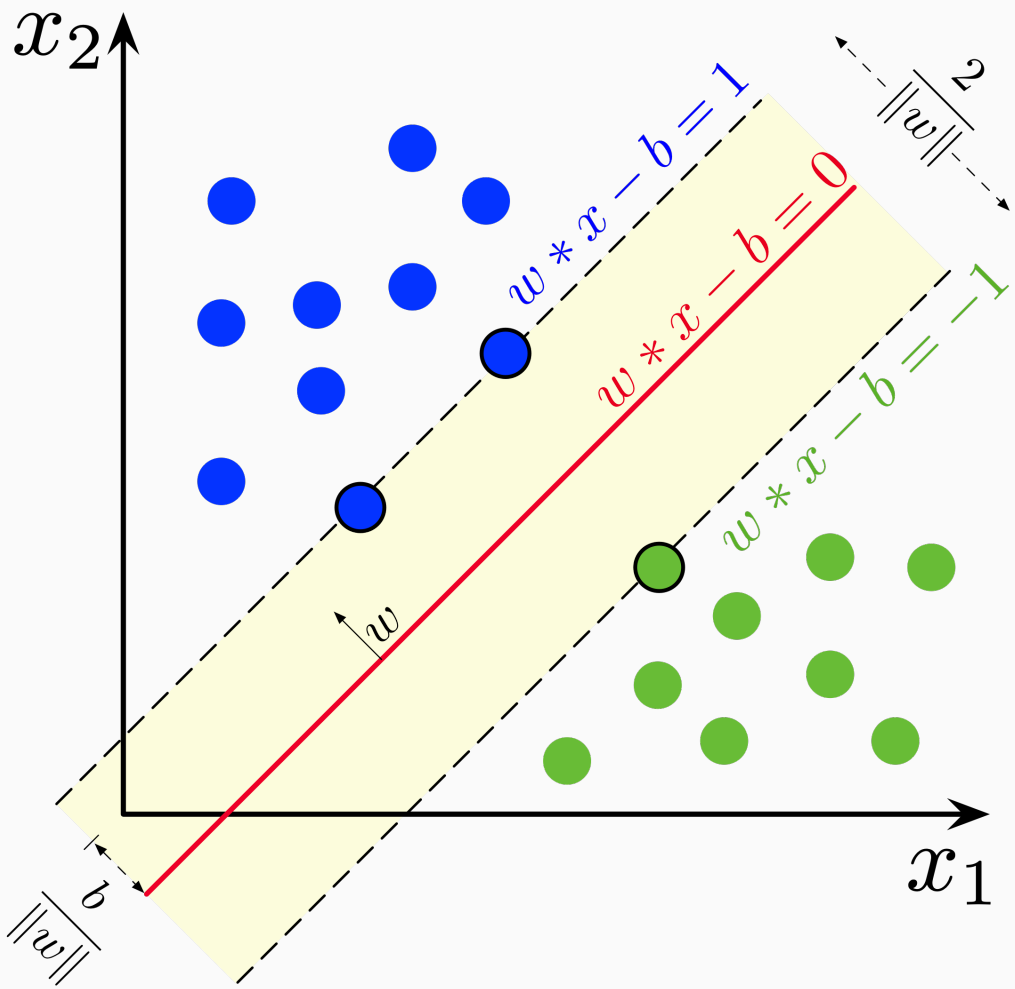

Figure 4: Support Vector Machine [Lar18]

TRADITIONAL MACHINE LEARNING TECHNIQUES

Traditional methods

- Linear, polinomial, logistic regression

TRADITIONAL MACHINE LEARNING TECHNIQUES

Traditional methods

- Linear, polinomial, logistic regression
- Decision Trees

TRADITIONAL MACHINE LEARNING TECHNIQUES

Traditional methods

- Linear, polinomial, logistic regression
- Decision Trees

  Gradient boosting (eg. XGBoost)

TRADITIONAL MACHINE LEARNING TECHNIQUES

Traditional methods

- Linear, polinomial, logistic regression
- Decision Trees

  Gradient boosting (eg. XGBoost)
- Support vector machine

TRADITIONAL MACHINE LEARNING TECHNIQUES

Traditional methods

- Linear, polinomial, logistic regression
- Decision Trees

  Gradient boosting (eg. XGBoost)
- Support vector machine
- Random Forest

TRADITIONAL MACHINE LEARNING TECHNIQUES

Traditional methods

- Linear, polinomial, logistic regression
- Decision Trees

  Gradient boosting (eg. XGBoost)
- Support vector machine
- Random Forest
- Genetic Algorithms

TRADITIONAL MACHINE LEARNING TECHNIQUES

- Traditional machine learning routines should be preferred over neural networks
- Deep learning can be powerful (and trendy)

  However, currently still limited in applicability

  Requires lots of data $\Rightarrow$ often not present
- Traditional Methods are faster to develop and test

  They typically expect same number of features with each datapoint

  $\Rightarrow$ Padding and Windowing are methods to circumvent this

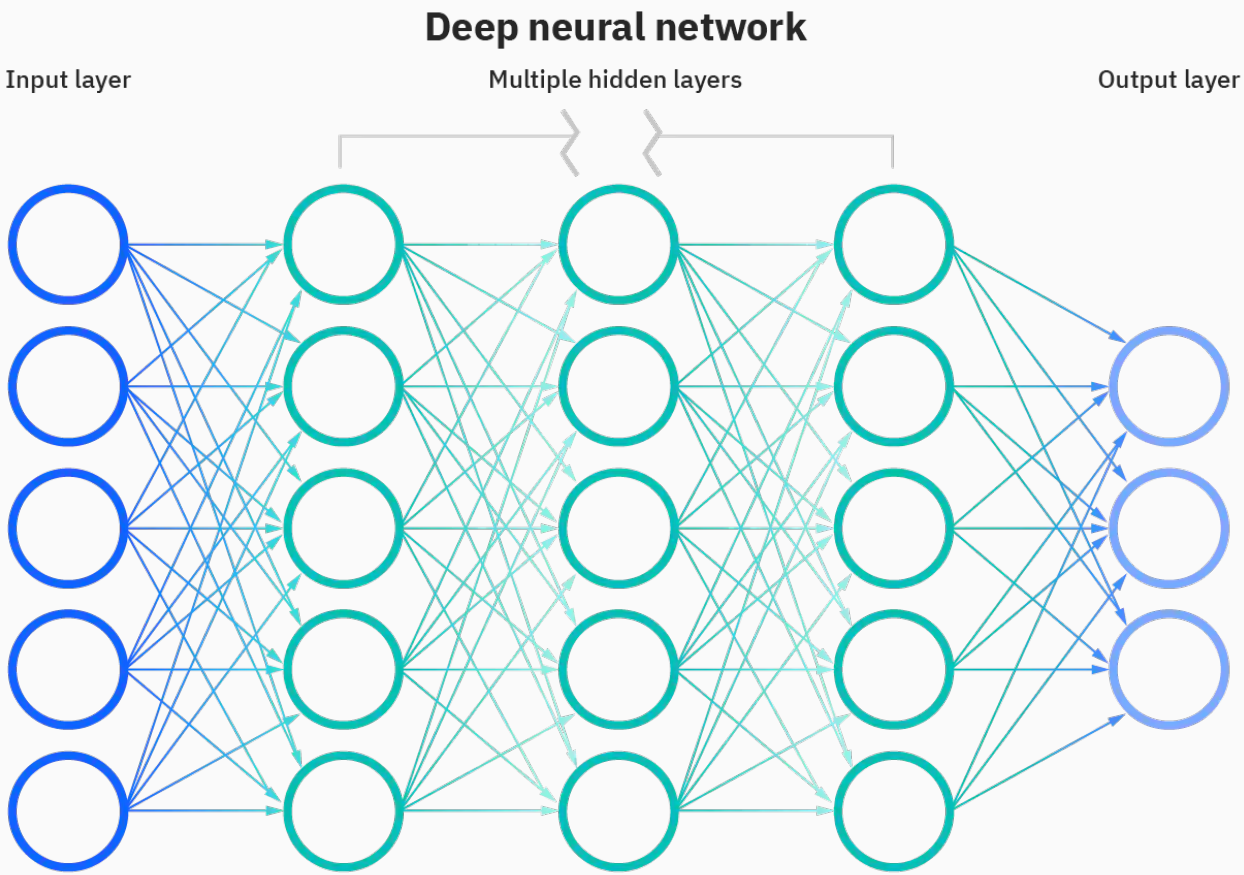

Figure 5: Standard, fully connected neural network [Edu20]

- Universal function approximators
- No guarantee that model will yield accurate predictins for new data
- Question: Is the trained model optimal?
- Neurons are at the heart of Neural Networks

  They apply a function to the input variables x_i to obtain output y by multiplying with learnable weight w_i.

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

- Multiple layers $\Rightarrow$ iterate this procedure

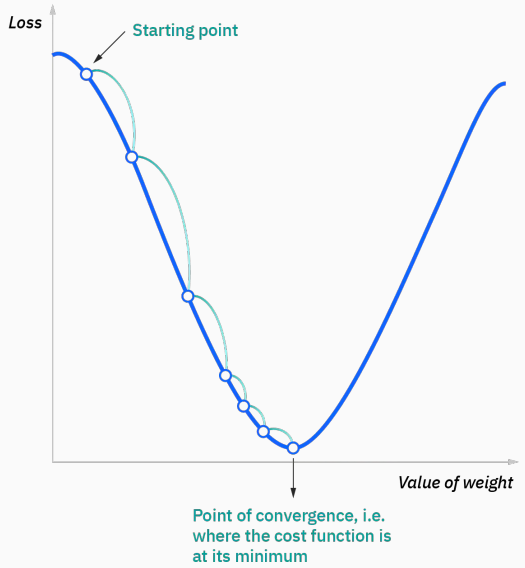

Figure 6: Iterative approach to finding a local optimum of the cost function. It penalizes or rewards good/bad results. An example can be given by the mean squared error [Edu20].

$$\text{MSE} = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2 \tag{2}$$

BACKPROPAGATION

1. Let $g(x_i)$ be a network output of input variables x_i.

BACKPROPAGATION

1. Let $g(x_i)$ be a network output of input variables x_i.
2. Let $C(y_i, g(x_i))$ be the loss function for predicted output $g(x_i)$ and target output y_i.

BACKPROPAGATION

1. Let $g(x_i)$ be a network output of input variables x_i.
2. Let $C(y_i, g(x_i))$ be the loss function for predicted output $g(x_i)$ and target output y_i.
3. Let $W^l = (w^l_{jk})$ be the weights between layer $l-1$ and l where w^l_{jk} is the weight between kth node in layer $l-1$ and jth node in l.

1. Let $g(x_i)$ be a network output of input variables x_i.
2. Let $C(y_i, g(x_i))$ be the loss function for predicted output $g(x_i)$ and target output y_i.
3. Let $W^l = (w^l_{jk})$ be the weights between layer $l-1$ and l where w^l_{jk} is the weight between kth node in layer $l-1$ and jth node in l.
4. Calculate the gradient of loss function in weight-space for fixed input-output pair (x_i, y_i)

$$\frac{\partial C}{\partial w^l_{jk}}(y_i, g(x_i)) \tag{3}$$

1. Let $g(x_i)$ be a network output of input variables x_i.
2. Let $C(y_i, g(x_i))$ be the loss function for predicted output $g(x_i)$ and target output y_i.
3. Let $W^l = (w^l_{jk})$ be the weights between layer $l-1$ and l where w^l_{jk} is the weight between kth node in layer $l-1$ and jth node in l.
4. Calculate the gradient of loss function in weight-space for fixed input-output pair (x_i, y_i)

$$\frac{\partial C}{\partial w^l_{jk}}(y_i, g(x_i)) \tag{3}$$

5. Multiple layers in Neural Networks mean that $\partial C/\partial w$ needs to be evaluated by chain rule.

1. Let $g(x_i)$ be a network output of input variables x_i.
2. Let $C(y_i, g(x_i))$ be the loss function for predicted output $g(x_i)$ and target output y_i.
3. Let $W^l = (w^l_{jk})$ be the weights between layer $l-1$ and l where w^l_{jk} is the weight between kth node in layer $l-1$ and jth node in l.
4. Calculate the gradient of loss function in weight-space for fixed input-output pair (x_i, y_i)

$$\frac{\partial C}{\partial w^l_{jk}}(y_i, g(x_i)) \tag{3}$$

5. Multiple layers in Neural Networks mean that $\partial C/\partial w$ needs to be evaluated by chain rule.

$\Rightarrow$ Modern Frameworks can do this process very efficiently.

1. Let $g(x_i)$ be a network output of input variables x_i.
2. Let $C(y_i, g(x_i))$ be the loss function for predicted output $g(x_i)$ and target output y_i.
3. Let $W^l = (w^l_{jk})$ be the weights between layer $l-1$ and l where w^l_{jk} is the weight between kth node in layer $l-1$ and jth node in l.
4. Calculate the gradient of loss function in weight-space for fixed input-output pair (x_i, y_i)

$$\frac{\partial C}{\partial w^l_{jk}}(y_i, g(x_i)) \tag{3}$$

5. Multiple layers in Neural Networks mean that $\partial C/\partial w$ needs to be evaluated by chain rule.

$\Rightarrow$ Modern Frameworks can do this process very efficiently.

6. To optimize: Go along steepest negative gradient of $\partial C/\partial w$.

BIOLOGICAL APPLICATIONS

BIOLOGICAL APPLICATIONS I

Input data	Example prediction tasks	Recommended models	Challenges
Gene sequence	DNA accessibility[14] 3D genome organization[58] Enhancer–promoter interactions[40]	1D CNNs RNNs Transformers	Repetitive regions in genome Sparse regions of interest Very long sequences
Protein sequence	Protein structure[23,55] Protein function[132] Protein–protein interaction[133]	2D CNNs and residual networks using co-variation data Multilayer perceptrons with windowing Transformers	Metagenome data stored in many places and therefore hard to access Data leakage (from homology) can make validation difficult
Protein 3D structure	Protein model refinement[134] Protein model quality assessment[135] Change in stability upon mutation[136]	GCNs using molecular graph 3D CNNs using coordinates Traditional methods using structural features Clustering	Lack of data, particularly on protein complexes Lack of data on disordered proteins
Gene expression	Intergenic interactions or co-expression[137] Organization of transcription machinery[138]	Clustering CNNs Autoencoders	Unclear link between co-expression and function High dimensionality High noise

Figure 7: Different applications in Biological contexts. I [GKMJ21]

BIOLOGICAL APPLICATIONS I

Mass spectrometry	Detecting peaks in spectra[139] Metabolite annotation[140]	CNNs using spectral data Traditional methods using derived features	Lack of standardized benchmarks[141] Normalization[a] required between different datasets
Images	Medical image recognition[24,62] Cryo-EM image reconstruction[60,142] RNA-sequencing profiles[143]	2D CNNs and residual networks Autoencoders Traditional methods using image features	Systematic differences in data collection affect prediction Hard to obtain large datasets of consistent data
Molecular structure	Antibiotic activity[73] Drug toxicity[54] Protein-ligand docking[39] Novel drug generation[144]	GCNs using molecular graph Traditional methods or multilayer perceptrons using molecular properties RNNs using text-based representations of molecular structure such as SMILES Autoencoders	Experimental data available for only a tiny fraction of possible small molecules
Protein–protein interaction network	Polypharmacology side effects[77] Protein function[145]	GCNs Graph embedding	Interaction networks can be incomplete Cellular location affects whether proteins interact High number of possible combinations

Figure 8: Different applications in Biological contexts. II [GKMJ21]

Questions?

[Edu20] IBM Cloud Education. Neural networks, 2020.

[GKMJ21] Joe G. Greener, Shaun M. Kandathil, Lewis Moffat, and David T. Jones. A guide to machine learning for biologists. Nature Reviews Molecular Cell Biology, 23(1):40–55, September 2021.

[HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527–1554, July 2006.

[Lar18] Larhmam. Support vector machine, 2018.

[Mew20] Sandeep Mewara. codeproject.com/, 2020.

[Mit97] Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.

[MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4):115–133, December 1943.

[Nil65] Nils J. Nilsson. Learning machines. McGraw-Hill, 1965.

[Skb09] Skbkekas. Polynomial regression, 2009.

[Tri20] Mayank Tripathi. Underfitting and overfitting in machine learning, 2020.