ML and DL*(udemy courses)*

What I have learned (in short):

Part 1:

- ***ML:***
  - *Classification models:*
    - Logistic Regression Model.
    - K-Nearest Neighbor Intuition.
    - Support Vector Machine(SVM) (Kernels SVM).
    - Naive Bayes.
    - Decision Tree.
    - Random Forest.
  - *Regression models:*
    - Simple(Multiple) Linear Intuition.
    - Polynomial Linear Intuition.
    - Support Vector for Regression (SVR).
    - Decision Tree Intuition.
    - Random Forest Intuition.
  - *Clustering models:*
    - K-Means.
    - Hierarchical:
      - Agglomerative.
      - Divisive *(just theory).*
  - *Association Rule Learning:*
    - Apriori Intuition.
    - Eclat Intuition *(just theory).*
  - *Reinforcement Learning:*
    - Upper Confidence Bound.
    - Thompson Sampling Intuition.
  - *Natural Language Processing(NLP).*
  - *Dimensionality Reduction Techniques:*
    - Feature Selection:
      - Backward Elimination.
      - Forward Selection *(just theory).*
      - Bidirectional Elimination *(just theory).*
      - Score Comparison*(just theory).*
    - Feature Extraction:
      - Principal Component Analysis(PCA).
      - Linear Discriminant Analysis(LDA).
      - Kernel PCA.
      - Quadratic Discriminant Analysis(QDA)*(just theory).*
  - *Model Selection Techniques:*

        - K-Fold Cross Validation.
        - Grid Search.
    - *XGBoost(just theory).*

NLP:

i. Bag of Words(A model used to preprocess the texts, to classify before fitting the classification algorithms on the observations containing the texts).
ii. Main library examples:
    1. Natural Language Toolkit – NLTK.
    2. SpaCy *(just theory).*
    3. Stanford NLP *(just theory).*
    4. OpenNLP *(just theory).*

Feature Selection*(Backward elimination, Forward selection, Bidirectional elimination)*:

i. Riding out from insignificant independent variables.
ii. Usage of P-value.

R-Squared Intuition(R^2):

i. R-Squared tells us how good is our line compared to our average line.
ii. Problem: when we add more variables R^2 will never decrease(we're adding variables and we will not know if those variables are helping our model or not).

Adjusted R-Squared Intuition:

i. Penalizes us for adding independent variables that don't help our model.

K-Means Clustering:

i. To define the optimal number of clusters; we use elbow method(usage of within cluster sum of squares(WCSS) algorithm).

Apriori:

i. Three parts for counting: *support, confidence, lift*.

Eclat:

i. There is a part: *support*.

Batch Learning(Off-line learning):

i. Update the weights only after a batch of observations.

Reinforcement Learning(On-line Learning):

i. Update the weights after each observation.
ii. Machines learn through trial and error.

Ensemble Learning:

i. When we take multiple algorithms or the same algorithm multiple times.

PCA:

i. Unsupervised algorithm.
ii. Goal: identify patterns in data(detect correlations between variables).
iii. For linear problem.

LDA:

i. We're looking for the class separation within the data(from the ‘n’ independent variables of our dataset, LDA extracts ‘p<=n’ new independent variables that separate the most the classes of the dependent variable).
ii. For *linear* problem.

Kernel PCA:

i. For *not linearly* separable data.

XGBoost:

i. The most powerful implementation of gradient boosting in terms of model performance and execution speed.
ii. A boosting model with decision trees.

K-Fold Cross Validation:

i. We choose the number of folds(for splitting our dataset in).
ii. For showing the most real results.

Grid Search:

i. We're finding the best *hyperparameters*.

Part 2:

- ***DL:***
  - *Artificial Neural Networks(ANN).*
  - *Convolutional Neural Networks(CNN).*
  - *Recurrent Neural Networks(RNN).*
  - *Self Organizing Maps(SOM).*
  - *Boltzmann Machines(BM):*
    - Restricted Boltzmann Machines(RBM).
    - Deep Belief Networks(DBN) *(just theory)*.
    - Deep Boltzmann Machines(DBM) *(just theory)*.
  - *Auto Encoders(AE):*
    - Stacked AE.
    - Deep AE *(just theory)*.

(ANN)(supervised method):

1. Input layer, hidden layer, output layer.
2. Not fully connected layers(it can be 'not connected' pairs of neurons between layers).

(CNN)(supervised method):

1. Convolution:
    i. (‘Input Image’ and ‘Filter’ = ‘Feature Map’).
2. Pooling:
    i. (‘Feature Map’ and ‘Pooling’ = ‘Pooled Feature Map’).
3. Flattening:
    i. (Array -> Vector).
4. Full Connection:
    i. Training similar to ANN.

(RNN)(supervised method):

1. Relationships:
    i. One to many.
    ii. Many to one.
    iii. Many to many(two forms).
2. The Vanishing Gradient Problem:
    i. Exploding Gradient:
        a. Truncated Backpropagation *(just theory)*.
        b. Penalties *(just theory)*.
        c. Gradient Clipping *(just theory)*.
    ii. Vanishing Gradient:
        a. Weight Initialization *(just theory)*.
        b. Echo State Networks *(just theory)*.
        c. Long Short-Term Memory Networks(LSTM).
3. LSTM:
    i. Standard.
    ii. Gated recurrent units(GRUs) *(just theory)*.
    iii. Other types *(just theory)*.

(SOM)(unsupervised method):

1. Retain topology of the input set.
2. Reveal correlations that are not easily identified.
3. No target vector -> no backpropagation.

(BM):

1. There is no output layer(we’re creating a model that describes our system).
2. Undirected model(no direction between nodes).
3. Generative type of model(capable of generating all of the values(it doesn’t need any inputs), it generates parameters on its own).
4. Boltzmann distribution:
    i. Than higher energy of a certain state, than lower is a probability.
    ii. Energy is defined in Boltzmann machines through the weights.
    iii. The system will always try to find the lowest energy state.

5. RBM:
    i. Visible and hidden nodes are not connected with each other.
    ii. Contrastive Divergence – the algorithm that allows us to find the weights for RBM.
6. Deep Belief Networks(DBN):
    i. Greedy layer wise training algorithm (algorithm for learning).
    ii. Wake-sleep algorithm (algorithm for learning).
    iii. There are directions in some layers.
7. Deep Boltzmann Machines (DBM):
    i. There are no directions.
    ii. They can extract features that are more sophisticated, more complex and therefore they could potentially be used for more complex tasks.

(AE)(self-supervised method):

1. Directed type.
2. Usage:
    i. Feature detection(hidden nodes will represent certain features).
    ii. Building powerful recommender systems.
    iii. Encoding.
3. Overcomplete Hidden Layers(when we want a greater number of HN(hidden nodes) than a number of IN(input nodes)):
    i. Sparse AE *(just theory)*.
    ii. Denoising AE *(just theory)*.
    iii. Contractive AE *(just theory)*.
4. Stacked AE:
    i. There are several hidden layers(some stages of encoding and one stage of decoding).

Image Augmentation:

i. This trick can reduce overfitting. This allows us to enrich our datasets or training sets without adding more images.

(Learning algorithms, Loss functions, Activation function, etc.)


Certificate of Completion
This is to certify that Valery successfully completed 23 hours of Deep Learning A-Z™: Hands-On Artificial Neural Networks online course on July 30, 2019
Kirill Eremenko
Kirill Eremenko, Instructor
Hadelin de Ponteves
Hadelin de Ponteves, Instructor
SuperDataScience Team
SuperDataScience Team, Instructor
&
Udemy
Certificate no: UC-ADAT2O74
Certificate url: ude.my/UC-ADAT2O74
#BeAble