**Artificial Intelligence - Vocabulary**

**Artificial Intelligence** – a computer system that show behavior in a way that could be interpreted as human intelligence, provided the system is dealing with pattern matching.

**Weak AI** – a computer system for only one tsk

**Strong AI** (Artificial General Intelligence) – displays a person like behavior

**Machine Learning** – A process in which machines start by identifying patterns, and then learn more as they work through additional data.

**Artificial neural network** – an artificial intelligence system that mimics the structure of the human brain. T

**Symbolic Reasoning** – a top down approach to AI. Involves extracting knowledge/symbols from a human and representing those in a computer-reable form

**Neural networks** - . This is considered a bottom up approach to AI: structure of a human brain, consisting of huge number of simple units called **neurons**. Each neuron acts like a weighted average of its inputs, and we can train a network of neurons to solve useful problems by providing **training data**.

**AI winter** – a period of reduced funding and interest in AI research (1970s)

**Alpha-Beta pruning** – a search algorithm that seeks to decrease the number of nodes in a search tree. Used in machine playing of two-player combinatorial games( Tic-tac-toe, Chess)

**Reinforcement learning** – a Machine Learning paradigm – a type of learning that sets goals and provides rewards

**Supervised learning** – a type of learning process in which a data scientist shows a system the correct answer and then allows the system to train itself to improve on identifying patterns that lead to correct answer.

**Unsupervised learning** – a type of learning process in which a system creates its own clusters of data based on observing patterns that it sees in the data.

**Eliza** -  an early natural language processing computer program created in 1960s Simulated conversation by using a pattern matching and substitution methodology that gave users an illusion of understanding the part of the program but really did not understand the context of conversation.

**Semantic network** – knowledge representation technique for organizing and storing knowledge. Semantic networks are a type of graphical model that shows the relationships between concepts, ideas, and objects in a way that is easy for humans to understand.

**Forward Inference** – process that starts with some initial data about the problem available in the working memory and then executes a reasoning loop based on a set of rules and conditions.

**Backward Inference** – process that starts with an empty knowledge about the problem, and ask questions that will help us arrive to the conclusion. It is driven by the goal - the attribute value that we are looking to find:

**Perceptron** - is a binary classification model, i.e. it can distinguish between two classes of input data.

**Cost function** – a measure of “wrongness” which in turn determines how much adjustment to weights and biases is needed

**Gradient descent** – used to change the weight and biases of artificial neural network – this will increase or decrease the cost function.

**Backpropogation** (backprop) – a method of training a multi-layers perceptron

**TensorFlow** (from Google) – provides a low level API to operate with tensors on both CPU and GPU ([TensorFlow](https://www.tensorflow.org/))

**PyTorch** (from Facebook)- provides a low level API to operate with tensors on both CPU and GPU ([PyTorch](https://pytorch.org/))

**Overfitting** - machine learning behavior that occurs when the machine learning model gives accurate predictions for training data but not for new data - occurs when the model cannot generalize and fits too closely to the training dataset – model maybe too powerful; may not have enough training data; too much noise in the input data.

**Underfitting** – when the model is not powerful enough

**Bias error** – algorithm not being able to capture the relationship between training data correctly

**Computer Vision** – aim is for computers to gain high level understanding of digital images - including finding an object on a picture (**object detection**), understanding what is happening (**event detection**), describing a picture in text, or reconstructing a scene in 3D.

**Convolutional Neural Networks(CNN)** - special type of neural networks used for computer vision. Sample CNN Architectures:

* VGG-16
* ResNet
* Google Inception
* MobileNet

**Transfer learning** – using a neural network trained on one dataset and adapt to classify different images without requiring a full training process: transferring some knowledge from one neural network model to another.

**Variation loss** – this is added into the loss function; is a metric that shows how similar neighboring pixels of an image are. Minimizing vrition loss makes an image smoother and gets rid of noise.

**Adversarial attacks** – [Generative Adversarial Networks (GANs) have been used to generate adversarial attacks, where two neural networks compete with each other](https://skimai.com/blog-what-is-an-adversarial-ai-attack/)

**Batch Normalization** – training technique that helps bring signals back to “normalcy”. Take into account ll value across a minibatch then normalize (subtract mean and divide by standard deviation). Ideally this will allow us to see a higher final accuracy and faster testing.

**Autoencoders** – a type of artificial neural network used to learn efficient codings of unlabeled data. Value provided by autoencoders: lower the dimension of images for visualization; denoising – removing noise from the mage; super-resolution – increasing image resolution; generative models.

**Encoder**- converts input image into some latent space

**Decoder** – reconstructs the original image

**Generative models** - models that can generate new images similar to the ones in the training dataset.

**Variational Autoencoders** – the autoencoder that learns to predict statistical distribution of the latent parameters. One important advantage of VAEs is that they allow us to generate new images relatively easily, because we know which distribution from which to sample latent vectors.

**Generative Adversarial Models – GANs** :  two neural networks that will be trained against each other.  adversarial because there is a constant competition between the generator and the discriminator. During this competition, both generator and discriminator improve, thus the network learns to produce better and better pictures.

* Generator is a network that takes some random vector, and produces the image as a result
* Discriminator is a network that takes an image, and it should tell whether it is a real image (from training dataset), or it was generated by a generator. It is essentially an image classifier.

**Segmentation** – precise object localization

* Semantic segmentation only tells the pixel class, and does not make a distinction between different objects of the same class
* Instance segmentation divides classes into different instances.

**Character-level representation** – (text representation) representing text by treating each character as a number.

Word-level representation - creating a vocabulary of all words in our text, and then represent words using one-hot encoding.

**Tokens -** converting text into a sequence of tokens, one token being either a character, a word, or sometimes even part of a word.

**N-grams** - could be comprised of large blocks of words, or smaller sets of syllables.

**Bag of words** - essentially represents which words appear in text and in which quantities, which can indeed be a good indication of what the text is about.

**Word2Vec** – a way to train semantic embeddings. based on two main architectures that are used to produce a distributed representation of words:

* Continuous bag-of-words (CBoW) — in this architecture, we train the model to predict a word from surrounding context.
* Continuous skip-gram is opposite to CBoW. The model uses surrounding window of context words to predict the current word.

**Language modeling** - creating models that somehow *understand* (or *represent*) the nature of the language. Semantic embeddings such as Word2Vec is a first step towards language modeling.

**Recurrent Neural Networks** – a type of artificial neural network that can process sequences of data; produce predictive results in sequential data that other algorithms can’t. Example – You’re reading a story; you read each word one by one; as you read you remember the previous word to make sense of the current word. RNN’s use their internal memory to process and make predictions based on the context of previous inputs.

**Vanishing gradients** – this is a problem with RNNs. RNNs are trained end-to-end in one backpropagation pass, it has difficulty propagating error to the first layers of the network, and thus the network cannot learn relationships between distant tokens.

**Sentence transduction** – this is any sequence to sequence tsk within machine translation. Within RNNs sequence-to-sequence is implemented by two recurrent networks, where one network, the encoder, collapses an input sequence into a hidden state, while another network, the decoder, unrolls this hidden state into a translated result.

**Attention Mechanisms** provide a means of weighting the contextual impact of each input vector on each output prediction of the RNN.

**Transformer Models** - a powerful class of neural networks that excel at analyzing sequential data by capturing long-range dependencies. Example is GPT-3

**BERT (Bidirectional Encoder Representations from Transformers**) is a very large multi-layer transformer network with 12 layers for *BERT-base*, and 24 for *BERT-large*. The model is first pre-trained on a large corpus of text data (WikiPedia + books) using unsupervised training (predicting masked words in a sentence).

**Named Entity Recognition (NER)** – a NLP task accomplished with neural networks; deals with recognizing specific entities within text, such as places, person names, date-time intervals, chemical formulae and so on.

**Token classification models** – NER models are bsicaly token classification models as for each of the input tokens we need to decide whether it belongs to an entity or not. We can use RNNs to train NERs – since NERs are essentially token classification model. Token classification models or token classification is very common in NLP – help to recognize specific entities within text including places, names, dates, and more.

**GPT: Generative Pre-Trained Transformer** - Large Language Models that are capable of solving tasks without any domain specific training.

**Perplexity** – measures the quality of the language generating model. This is an intrinsic metric that allows the ability to measure the nodel quality without any taskspecific dataset. – based on the probability of a sentence.

**Prompt engineering** - GPT inputs or queries whereby one provides instructions to models on tasks they next completed. To elicit a desired outcome, you need the most effective prompt which involves selecting the right words, formats, phrases or even symbols.

**Reinforcement learning (RL)** is seen as one of the basic machine learning paradigms, next to supervised learning and unsupervised learning. RL is based on learning by doing. To perform RL you need an environment/simulator that sets the rules of the game AND some sort of reward function. The goal of a RL algorithm is to train a model which will return the action in response to a given state.

**Actor-Critic Algorithm**. The main idea behind it is that the neural network would be trained to return two things:

* The policy, which determines which action to take. This part is called actor
* The estimation of the total reward we can expect to get at this state - this part is called critic.
* this architecture resembles a [GAN](https://microsoft.github.io/AI-For-Beginners/lessons/4-ComputerVision/10-GANs/README), where we have two networks that are trained against each other.

**Principles of Responsible AI**

**Fairness** is related to the important problem of model biases, which can be caused by using biased data for training

**Reliability and Safety**. AI models can make mistakes. A neural network returns probabilities, and we need to take it into account when making decisions.

**Privacy and Security** -when we use some data for training a model, this data becomes somehow "integrated" into the model. While this increases security and privacy, we need to remember which data the model was trained on.

**Inclusiveness** - we are not building AI to replace people, but rather to augment people and make our work more creative.

**Transparency** - making sure that we are always clear about AI being used.

**Accountability** - make sure that we understand where responsibility of AI decisions lies: want to include human beings into the loop of making important decisions, so that actual people are made accountable.