EYE4AI

WHAT IS AI?

EMPOWERING YOUNG ENTREPRENEURS WITH AI

Funded by the European Union

n° 2024-1-BG01-KA220-YOU-000249027

CONTENTS

Funded by the European Union

INTRODUCTION

In this module the fundamentals of AI are discussed, providing information about the pillars needed to create an AI model.
The bottlenecks of creating powerful AI models are discussed, which include design decisions on an architectural level as well as nuances which need to be considered when training models.

After reading this module you should have a deeper understanding of the themes related to AI. This module focuses mainly on the technological side of AI but also suggests how to most effectively use AI.

- You will learn about how significant training data is and what to look out for.
- You will learn about the different architectures that are available currently as well as their strengths and weaknesses.
- You will learn about the most productive ways to use currently available AI models.

Funded by the European Union

1

AI AND MACHINE LEARNING

As the industry has evolved, the perception of the meaning of AI has changed. In the modern day when people say AI they are primarily referring to large machine learning models, which is what will be discussed in this module. The concept of machine learning refers to an algorithm that can find patterns in input data and improves itself as it is trained more extensively. The key is that it is not told how to think. Some models work by mimicking the biological brain.

In some cases it is just a buzzword used to grab attention and sell products so it is crucial to understand the inner workings of this mysterious and often misunderstood technology. By understanding the key concepts you will have a better idea of what is possible and not be misled by hype. You will also be better equipped to take advantage of the power that AI has to offer.

In this module we will cover the understanding of key topics in machine learning. We will highlight the importance of high quality training data. Then we will cover hand picked examples of AI architectures since there are so many, including information about how they work and what to expect from them as strengths and weaknesses. Consequently we will look into how different AI models can be trained and the types of training, therefore blending the architecture and training data. Finally we will look at the current state of AI and comment about how to get the most out of the products available on the market.

Funded by the European Union

2

IMPORTANCE OF TRAINING DATA

As in most systems, AI is only as strong as its weakest link. If you have good data but no way to process it you have nothing, and likewise even if you have the most advanced architecture but you feed it low quality data the results will be underwhelming.

In this chapter we will discuss the importance of providing high quality data when training an AI model in addition to evaluating what criteria data needs to meet to be considered high quality.

There are systems that use simulated learning where an AI interacts with a simulated environment and learns by trial and error through reinforcement (this will be explained in chapter 4). For other AI models training data is the only source of understanding for the AI. All of the patterns observed by the model are based on the data it is being fed. This is because the AI, unlike humans, has no body or personal experience to draw knowledge from and is reliant on the input to define its truth. It has no further way to test its conclusions unless it is later told that the information is incorrect. This is why high quality data is necessary for AI models to function. That being said, what constitutes high quality data?

The first consideration is accuracy. While there are both labeled and unlabeled sets of data, accuracy is critical to both. When using labeled data in combination with supervised learning (more on this in chapter 4) the accuracy of the data relates to how detailed and correct the labels are relative to the data. For example you can train an AI to identify objects in an image

EXAMPLE INPUT IMAGE
RSPB.ORG.UK/BIRDS-AND-WILDLIFE/ROBIN

Take this image for example, it may be used as an input to train an AI. The image stays the same but the tags may vary depending on the desired complexity of the AI. One label may be “bird”; this will limit the understanding of the AI. Another more advanced dataset may label this image with tags “bird, robin, perched, standing on branch, orange chest, blurred background” and so on. You can imagine the difference in the resulting AI models. If you use an unreliable dataset this image might even have a label “cat”. Of course this is exaggerated, but if you teach the AI that this is a cat, even if it understands what a bird is it might refer to it as a cat. For this reason using detailed and accurate labeled data is critical for proper functionality when training an AI.

If we want to give an example for accuracy in the context of unlabeled data used in unsupervised training, we can imagine a large language model (LLM). LLMs such as ChatGPT can be trained on unlabeled data such as articles, tweets, reddit posts, science journals, etc. This is unlabeled data

Funded by the European Union

and the AI uses it to identify patterns in the sentence structure and forms relationships between words. By training it on enough of this data the AI starts to produce sentences that a human might write. If, however, you train it on school homework written by children of age 6, you may get undesired outputs from the AI. This is what quality means in the context of unlabeled data.

Shortly put, if you teach it things that aren't true the AI has no way to correct this. Make sure the data being used to train an AI is high quality or it might identify cats as dogs. If you are training it on text, make sure it is written in coherent and grammatically correct text otherwise the outputs may be nonsensical. Even if a small subset of the data is low quality it may contaminate the model since AI works on patterns and it will make incorrect connections. The quality of the input data is directly correlated to the quality of the outputs.

Another consideration for high quality data is built in biases. Even if there is technically nothing wrong with the data you can still have massive problems if you overlook biases. In one PR scandal an image recognition AI seems to have been trained mostly on caucasian human faces which lead to it identifying people from other races as primates. This can be solved by taking into consideration the 2 next criteria for high quality data - volume and diversity.

The larger the brain of the AI - measured in number of parameters - the more data is needed to accurately and effectively encode values into its nodes since each input will only slightly modify the node values. Think of nodes as mathematical functions - they take some input and give some output after doing some math. For this reason, it is important to have sufficient volume when training an AI. There are various suggestions online for how much data is enough, but the only real way to tell is by testing the AI. This of course tests the whole system and not only the amount of input data so it is difficult to identify the weak point, but by testing the model you will get a feel for the results you can expect.

The final important consideration for training data is the diversity. If you give the AI too much of one type of input it will overfit and not recognize general patterns, but only be optimised for that one question.

Of course this depends on the purpose of the AI but for large language models it is critical that they have a broad understanding and try to reason on their own. Let's see as some examples: if you only show it adding questions then it will not know how to multiply. If you show it too many articles then it will sound like a reporter in all contexts so you may want to include reddit threads or tweets. In an image recognition AI you may feed it too many similar photos - if all photos with a bird have a blue sky background then it may assume that the blue is contributing to the image being a bird.

3

AI ARCHITECTURES

While training data is important for the final functionality of an AI model, it is not the only consideration. In order for that data to end up being useful an appropriate architecture to base the model on is crucial. Think of the architecture as a skeleton, just a bunch of bones which need to be connected by muscles. There are many types of architecture that can be used for different purposes. Some AI models are narrow, known as artificial narrow intelligence (ANI) and are optimized for one very specific purpose like chess AIs, while others are called artificial general intelligence (AGI) and attempt to mimic human understanding in all fields. AGI has not been achieved yet and there are many debates about definitions and consciousness. Large language models (LLMs) like ChatGPT are an attempt at creating AGI and are significantly more advanced than the technology was a few years ago. That being said, for either category there are many different architectures, each with their superpowers, limitations, and specific use cases for which these tradeoffs make sense. To give a basic understanding of what is possible we will only discuss 2 of these - Genetic Algorithms and Neural Networks.

3.1: Genetic Algorithms

Genetic algorithms (GAs) are one type of AI architecture, a little different than what you are probably used to seeing in the mainstream. In order to widen understanding of different types of AI we will look into what they are useful for and how they work. This is a very specialized type of architecture that is not broadly useful for many different types of tasks but is highly efficient for particular problems. Usually GAs are used in optimization problems where there are more variables than what can be brute forced. It is an elegant lightweight solution for testing the possible solutions to strictly defined problems. All of this may sound a little abstract so let's investigate the most popular example - the traveling salesman problem.

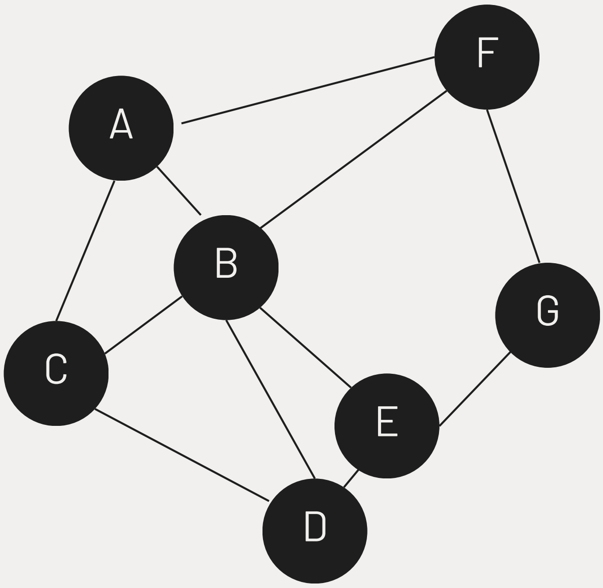


DIAGRAM OF POTENTIAL SCENARIO

In the drawing above you can see 7 nodes. The traveling salesman problem expects a solution where every node or city is visited exactly once, and the goal is to find the path that is shortest. For simplicity there are no distances between the nodes in this example. One potential solution is ABCDEFG. Another solution is ABCDEGF. While there are other ways to solve this, GAs perform really well in such problems.

So how do they work? In this example, every solution is 7 letters long and includes each letter exactly once. Think of this as the DNA on which the genetics are based, the location of each letter is an attribute of the solution. Each solution is a member of the population.

Initially, a random population is generated with all types of solutions, some of which are inefficient and others are faster. A fitness function is a ranking system used to compare the quality of solutions. In this example the fitness function is just the distance of the path with lower distance being better. This means that lower total distance solutions will be ranked better and be used to produce the next generation. With 3 nodes if the solution is ABC, the distance is AB + AC, where AB is the distance from A to B.

Once the population is generated, it should be sorted on the fitness function. Then, the next generation can be produced from parents of the current generation. Although higher ranked solutions are more likely to reproduce, there is some randomness to avoid premature convergence and local maxima. This topic is too nuanced for a deep dive within the scope of this course but extra resources can be found for curious learners.

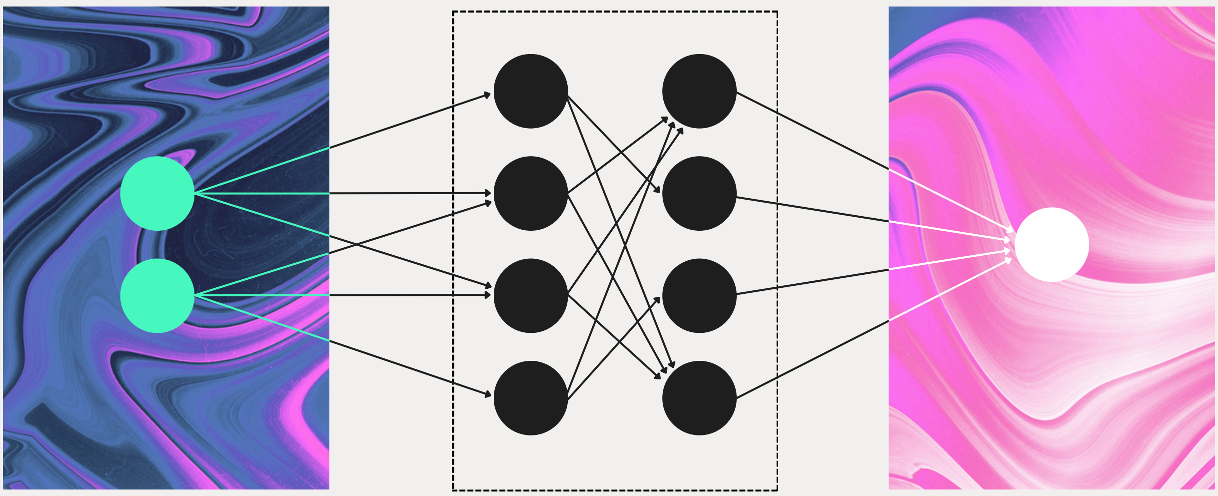
Reproducing is done by taking the 2 parents and combining them to vary the offspring. While there are many different ways to combine the genetics, the general idea is that the new population is on average better since the least effective solutions from each generation are removed. By running the algorithm for multiple generations you can keep

3.2: Neural Networks

Neural networks (NNs) are the cornerstone of modern AI. The concept is that they mimic how human brains work with neurons (brain cells) and the connections between them. In the context of AI each neuron is called a node and its computer representation is a function for processing data. One simple case is where the node has a value between 1 and 0 and the input is multiplied by the value, this is a function. The connections also frequently have values called weights, representing the influence of one node on the next.

There are a vast number of different structures for NNs, including the transformer architecture - this is what most LLMs like ChatGPT are based on. In this chapter we will discuss the broad idea of a neural net rather than the details that make up more advanced architectures.

To develop an intuition for how NNs work we will use a visual example of an image recognition AI.
In this example the nodes are the circles and the arrows are the connections. The weights of the connections and the nodes both impact the results. We will understand where these values come from in chapter 4 - training AI.



ARTIFICIAL NEURAL NETWORKS; DIAGRAM OF POTENTIAL SCENARIO
HTTPS://WWW.SCIENCELEARN.ORG.NZ/IMAGES/5156-NEURAL-NETWORK-DIAGRAM

Funded by
the European Union

Now imagine a grid like this over an image. Instead of these squares, however, each pixel is a blue input circle in the NN. These inputs are then processed by the network, each connection analysing patterns between the neighbouring pixels. When the input passes through enough hidden layers it is capable of identifying structures based on the input pixels. It would then be able to describe what these structures are.

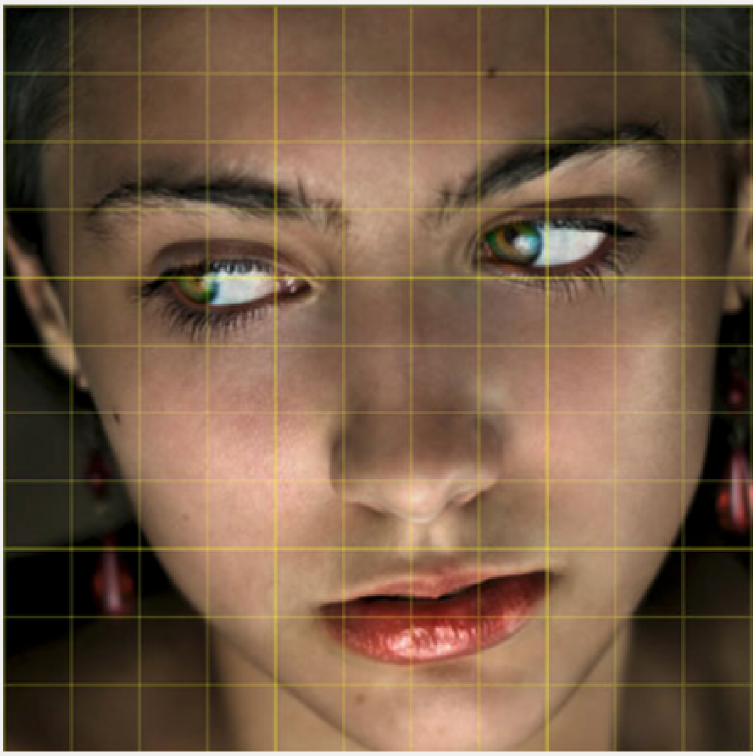


IMAGE WITH GRID
WWW.PHOTOSHOPESSENTIALS.COM/PHOTO-EFFECTS/PHOTO-STRIPS/

The capabilities are dependent on the size of the neural net as it can only take as many inputs as it has nodes. Think of it as a puzzle: if each node could tell you where a piece goes, you would need as many nodes as you have pieces. This is oversimplified though and in reality it is beneficial to have more nodes for accuracy and nuance. While we learned that training data is important in the previous chapter, the size of the NN is also a bottleneck for performance. If you had one node, no matter how much data you fed it and how good your data is, all you would do is update one value and in the end your AI would not be useful.

With the understanding from this chapter it should be clear that the architecture of an AI model will directly impact what it is capable of. Neural networks are more versatile as the input data can be flexible if the size of the brain (number of parameters) is sufficient. Genetic algorithms are use case specific due to the enforced input and output, but certainly have their strengths when used in the right context. There are far more architectures to be explored, some very advanced and some more basic. In general basic models are faster to run and preferred for well defined expectations. Advanced models are expensive to train and require state-of-the-art hardware to run, but produce mindblowing results if engineered effectively. Always make sure to analyse which architecture is suitable as it will determine the results. Stockfish is a good chess AI, better than any human in history, while ChatGPT still makes illegal moves throughout its games. Both are AI but they are trained with different intentions.

4

TRAINING AI

Now that we have covered the importance of selecting the right training data and architecture, it is time to see how it all comes together. The final step is the process of training the AI. You built the building, you bought the furniture, and now it is time to arrange it. In this chapter we will discuss several relevant topics regarding training AI. In the context of GAs we will explain the importance of fitness functions and also relate this more broadly to reinforcement learning. After that we will look into the difference between supervised and unsupervised learning.

4.1: Fitness Functions and Reinforcement Learning

While there are some differences between these 2 concepts, they have been grouped together since the AI learns from interacting with an environment. Fitness functions are prominent in GA as the solutions need to be ranked in some way so the parents can be selected. It is therefore important to consider what exactly the most effective fitness function is.

Let's pretend for a moment that Google maps used an AI for finding the shortest path. What would the fitness function be? Would it be the shortest distance? Least amount of fuel consumption? Least amount of time to get to the destination? There are always trade-offs - it is rather simple to calculate the distance, but if you want to calculate the time you need to consider the maximum speed of each road, traffic lights, road traffic, road closures, etc. This might take longer for the system to score the solutions and slow down the evolution process. It is important to have an appropriate fitness function, however, as it determines the results you will get. The answers for fastest path and fuel efficiency might even be different.

For example if you are training a chess AI and you tell it the number of pieces on the board are the fitness function, you might get checkmated despite not losing any pieces. That is why it needs to be carefully considered what fitness function to use.

Reinforcement learning is related in some ways to this as it includes fitness functions. It is prominent in simulated learning - where the AI is in a simulated environment and is attempting to achieve some goal. The concept of reinforcement learning is that the AI gets rewarded for doing the right thing and punished for doing the wrong thing. For example if the AI is a car in a racing game, it gets more points for reaching the finish line faster, it may also lose points for going in the wrong direction. In this way, every next generation the better solutions are kept and the worst ones do not reproduce. Once again the function used to score the AI should be carefully crafted.

The AI should be given enough room to figure things out as it may find a better way than you anticipate, but if you are too strict with the fitness function then it will find the solution you want which you can find yourself. In the same example of the racing game, if you give it points for going tightly around the corners you are being strict and not allowing it to experiment. It may be the case that going far from one corner and maintaining speed is more effective in the context of the whole race.

4.2: Supervised and Unsupervised Learning

GAs, fitness functions, and reinforcement learning are all related since they use interactions with an environment to train the AI through telling it what is scored higher as a result. The other paradigm is not telling the AI whether it is doing well but rather just train it on a bunch of data and see what happens.

In supervised learning, despite the name, there is no human directly monitoring the training process.

Supervised learning is when the data is labeled (usually by humans). This means that someone has provided the “correct answers,” much like studying for an exam by using the solutions to previous exam questions. This is beneficial when there is a clear result that is expected, and it produces AI models which are more predictable and less volatile (have less variance in accuracy and consistency).

Unsupervised learning is when the data is not labeled, and the AI scans for patterns within the data itself rather than making connections between the data and the labels. This is useful for when the expectations are more subjective, like in written English - there are many ways to say the same thing and none of them are inherently wrong. In this case it is impractical to label the data and is more effective to allow the AI to discover patterns on its own.

Now that we know the distinction between supervised and unsupervised learning is the type of data being given to the AI, let’s consider the applications of both. In advanced models like LLMs (ChatGPT) both are used to give the AI more capability, but what is each useful for?

The distinction is important as either type of learning comes with particular traits. Supervised learning may be expensive as it requires the data to be labeled - this is an intensive process due to the volume of data needed to train a complex model. It comes with the advantage that the AI will know the expected output as it is well defined and clearly structured. Unsupervised learning, on the other hand, can use any data you give it and try to construct patterns from that. It therefore is less explicit and structured in the results but is easier to train as it does not require special data. It is good at finding small nuances and patterns that are otherwise not obvious.

Funded by
the European Union

5

HOW TO USE AI EFFECTIVELY

Now that you have developed an understanding for what goes into creating an AI model it is time to look into the current landscape of existing products and how they can most effectively be used. It is no secret that using AI can supercharge your productivity so in this chapter we will discuss the best uses for LLMs and cover in which situations AI may be counterproductive. LLMs like ChatGPT can be useful for many situations, but since this course is for young entrepreneurs we will focus on the most business oriented uses.

BRAINSTORMING

BENEFITS

- Fast generation of related topics
- Familiar with many concepts and good at linking related points
- Can bounce ideas off you, similar to how you would talk to a person

LIMITATIONS

- Sometimes does not understand what you are expecting so you need to be more specific
- Only trained on available data so if it is something proprietary or outside of the dataset you are out of luck.

WRITING COPY

BENEFITS

- Usually writes more coherently than most humans, especially in their 2nd language
- Familiar with many concepts and good at linking related points
- Can bounce ideas off you, similar to how you would talk to a person

LIMITATIONS

- Sometimes the tone is not relevant for the type of text
- If you are working with someone that does not accept AI content they can use an AI scanner.
- Might take longer to explain the context and expectation than writing it yourself

Funded by
the European Union

GIVING FEEDBACK

BENEFITS

- Spots mistakes that you may overlook
- Faster than you reading your own work, especially for larger texts
- Can suggest improvements once critiques have been identified

LIMITATIONS

- Programmed to give a response no matter how forced, resulting in sometimes trivial feedback
- Sometimes much more critical than necessary for the task at hand
- Often uses more words than necessary for the message it is trying to convey unless you instruct it not to

CODING

BENEFITS

- Fast code generation
- Familiar with many languages and frameworks
- Can fix its own code if it doesn't work first try
- Can explain the code line by line

LIMITATIONS

- Unable to generate file systems and complex integrations
- May waste time by generating code that you do not understand

SOME GENERAL CONSIDERATIONS WHEN USING AI

- Sometimes hallucinates false facts
- Limited to training data that has been provided
- Only knows the contextual information that you provide

LLMs work on the basis of predicting the next word that should be generated. While newer models have more advanced systems for citation and providing information, you still want to double check important facts yourself as mistakes can happen.
If the topics that you are trying to get information about or discuss with the AI are not prominent in its training data it will not do a good job in any prompts related to these topics.
If you are, for example, trying to write a text for the website of your company then you need to provide the AI with a whole lot of information about your vision, operations, and so on. If this information is not already in accessible text that you can copy paste, it may take longer for you to write it up than writing the website copy yourself. This is especially true when you need a short text with very precise meaning. In those cases it may be effective to write the initial text and then ask the AI if it sees any room for improvements.

Funded by
the European Union

6

RECAP OF KEY TAKEAWAYS

You have reached the end of the module, congratulations! Now let's go over some of the key takeaways so you don't forget the most important parts.
From chapter 2 we learned why training data has a significant impact on the outputs from an AI model. We also learned that high quality data should:

- Be accurate
- Have sufficient quantity
- Be diverse
- Consider unintended biases

Then in chapter 3 we covered how the type of architecture you chose is related to the type of problem being solved. We discussed that genetic algorithms are efficient for problems with a defined result format, especially optimisation problems. We also developed a basic intuition for how neural networks (NN) function and how there are many types of NN. The NNs are flexible in their purpose and functionality.
In chapter 4 we learned how reinforcement learning and fitness functions are related and how they work and what their purpose is. We also made the distinction between supervised and unsupervised learning.
Finally in chapter 5 we talked about how to best use AI and the considerations that may improve workflow. We outlined how AI performs in tasks like coding, brainstorming, writing copy, and giving feedback, along with its strengths and weaknesses in each category.
Hopefully this understanding is beneficial in your journey with AI, good luck.

BIBLIOGRAPHY

1. oracle.com/nl/artificial-intelligence/ai-model-training/#role-of-data
2. nytimes.com/2021/09/03/technology/facebook-ai-race-primates.html
3. eeksforgeeks.org/genetic-algorithms/

Funded by
the European Union

EYE4AI