week #6

classifier models



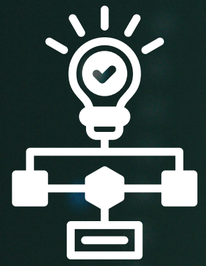
Presented by *Sergi, Aren and Michal*

roadmap



WEEK 1-5 :

- Fundamentals
- LLMs
- Diffusion
- Agents
- Ethics



NOW :

- Building Classifier Models
- Testing and Training



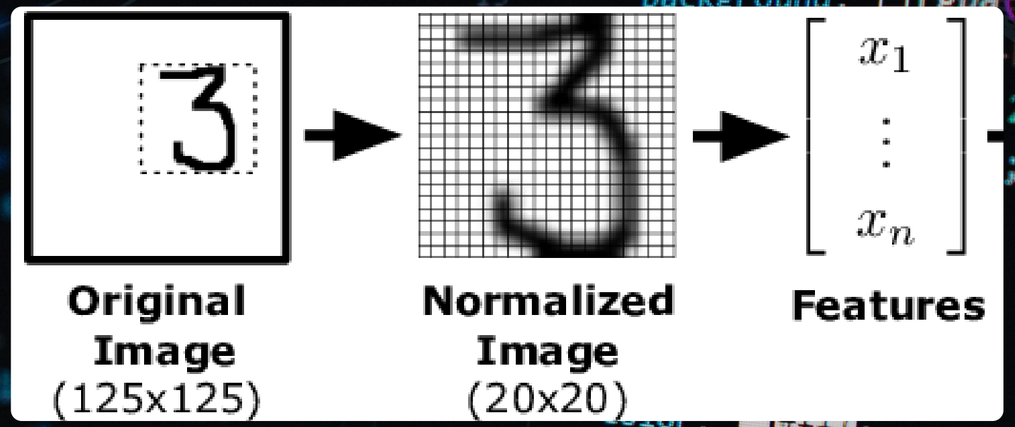
WEEK 7+ :

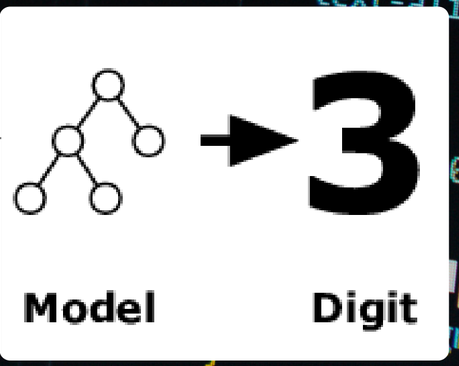
Real-World AI Showcases & New Developments

today's objective: computer vision

We are moving from theory to practice. Today, we will build a Neural Network that can "see" and "read" handwritten numbers.

- The Problem: Computers see images as a grid of numbers (pixels), not shapes.
- The Solution: A Machine Learning model that learns to recognize patterns (curves, loops, lines).
- The Goal: Train a model to look at a messy handwritten digit and correctly identify it as 0-9.

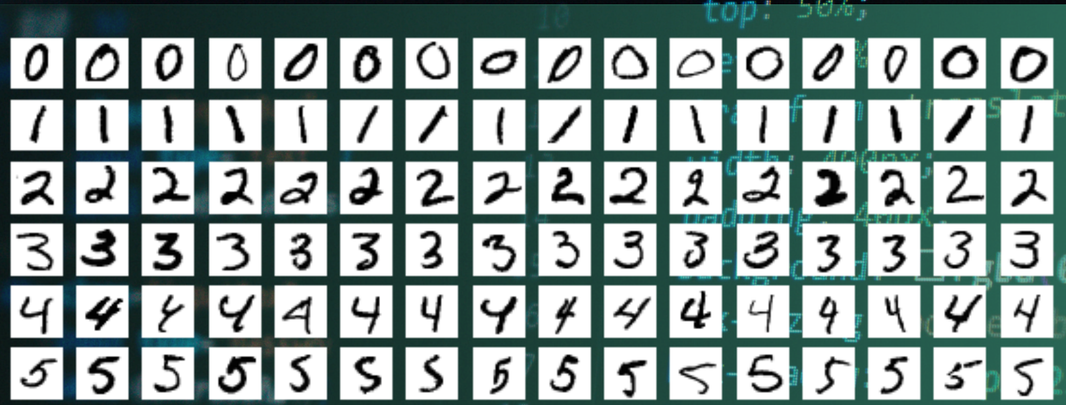


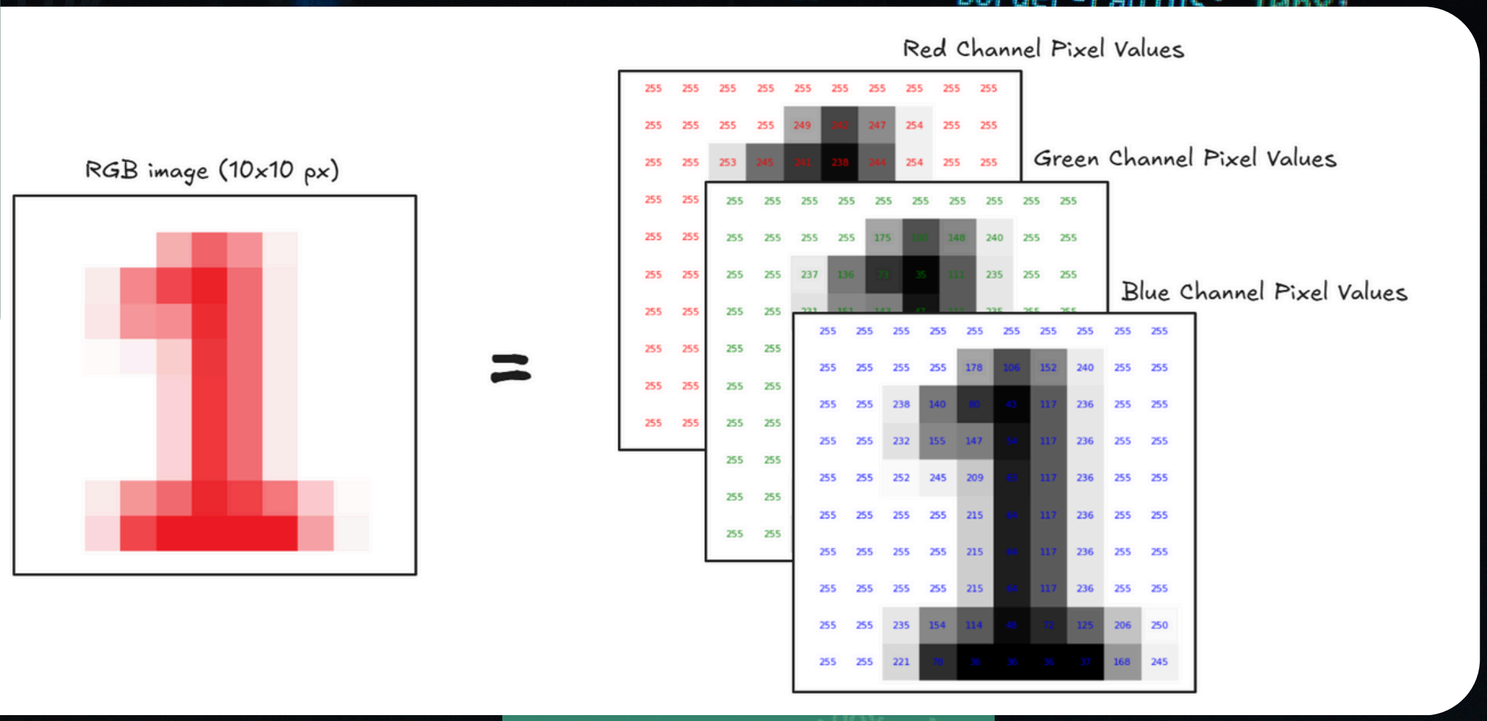


the training data

To teach the computer, we need examples. We will use the MNIST Dataset, the "Hello World" of AI.

- 60,000 Training Images: Handwriting samples from different people.
- Format: 28x28 pixel grayscale images.
- Normalization: We convert pixel brightness (0-255) into simple values between 0 and 1 so the math is easier for the AI to handle.





building the brain

```
[15]
✓ 0s
```

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 500)	392,500
dense_1 (Dense)	(None, 300)	150,300
dense_2 (Dense)	(None, 10)	3,010

Total params: 545,810 (2.08 MB)
Trainable params: 545,810 (2.08 MB)
Non-trainable params: 0 (0.00 B)

```
model = tf.keras.Sequential([
                    tf.keras.layers.Flatten(input_shape=(28,28)),
                    tf.keras.layers.Dense(500, activation='relu'),
                    tf.keras.layers.Dense(300, activation='relu'),
                    tf.keras.layers.Dense(10, activation= 'softmax')
                    ])
```

We will construct a Neural Network using TensorFlow (Google's AI library).

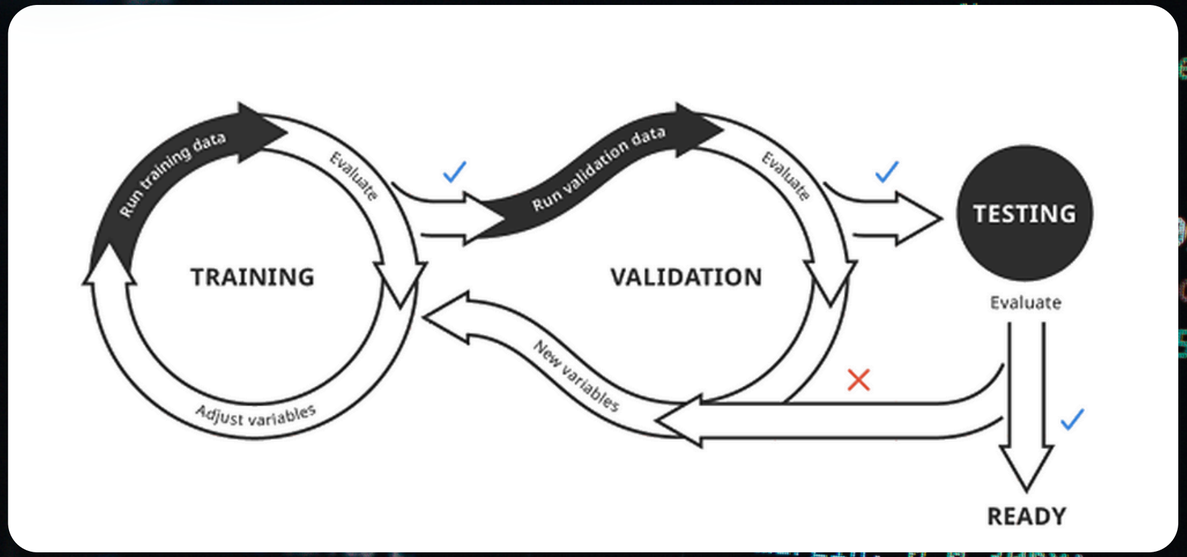
- Input Layer (Flatten): Takes the 2D image square and turns it into a long line of pixels.
- Hidden Layers (Dense): This is where the "thinking" happens. The nodes look for specific features like edges or circles.
- Activation Function: We use ReLU (Rectified Linear Unit) to help the model learn complex patterns.
- Output Layer (Softmax): Gives us a probability score for every possible number (0-9).

training: the learning loop

We don't program the rules; the model learns them by trial and error.

- Epochs: The number of times the model reads through the entire dataset.
- The Optimizer: Adjusts the "weights" of the neurons to reduce errors.
- Loss vs. Accuracy:

- **Loss:** How wrong the model is (we want this LOW).
- **Accuracy:** How often it guesses right (we want this HIGH).

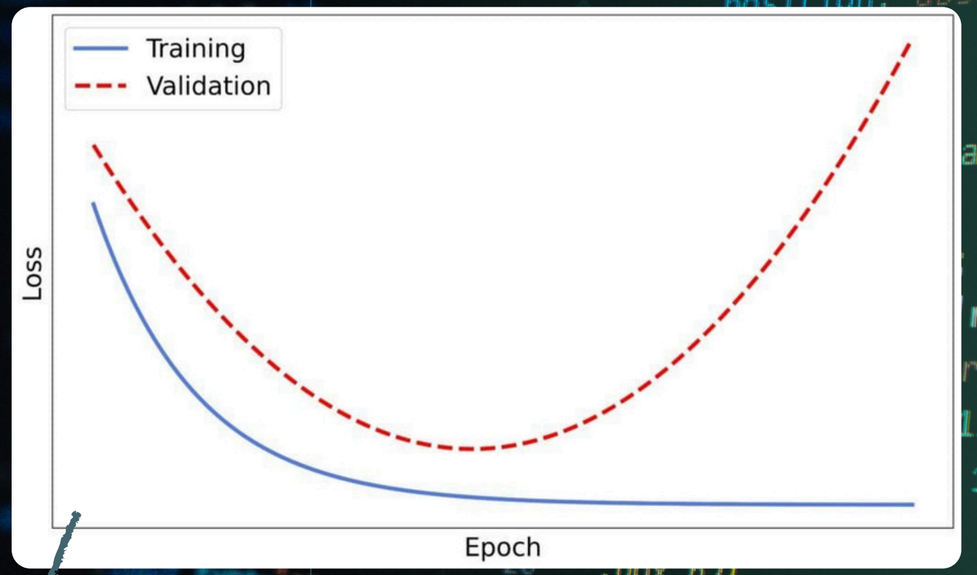



```
Epoch 1/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 13s 18ms/step - accuracy: 0.8863 - loss: 0.3948 - val_accuracy: 0.9704 - val
Epoch 2/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 11s 18ms/step - accuracy: 0.9750 - loss: 0.0813 - val_accuracy: 0.9750 - val
Epoch 3/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 8s 14ms/step - accuracy: 0.9845 - loss: 0.0501 - val_accuracy: 0.9739 - val_
Epoch 4/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 10s 16ms/step - accuracy: 0.9894 - loss: 0.0354 - val_accuracy: 0.9760 - val
Epoch 5/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 11s 18ms/step - accuracy: 0.9917 - loss: 0.0267 - val_accuracy: 0.9788 - val
Epoch 6/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 11s 18ms/step - accuracy: 0.9933 - loss: 0.0195 - val_accuracy: 0.9823 - val
Epoch 7/10
600/600 ━━━━━━━━━━━━━━━━━━━━ 8s 13ms/step - accuracy: 0.9946 - loss: 0.0174 - val_accuracy: 0.9798 - val_
```

testing & prediction

Once trained, we must test the model on data it has never seen before.

- Generalization: A model that memorizes the training data but fails on new data is "Overfitted."
- Confidence Score: The model doesn't just say "It's a 7." It says, "I am 98% sure this is a 7."
- Real World Test: We will upload our own drawings to see if the AI can read our handwriting.



General trend of loss decrease with epoch increase vs validation overfitting and model breakdown

```
predict_image('https://dojo.soy/num2')

Downloading data from https://dojo.soy/num2
4602/4602 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
Your model predicts this number is 2
```

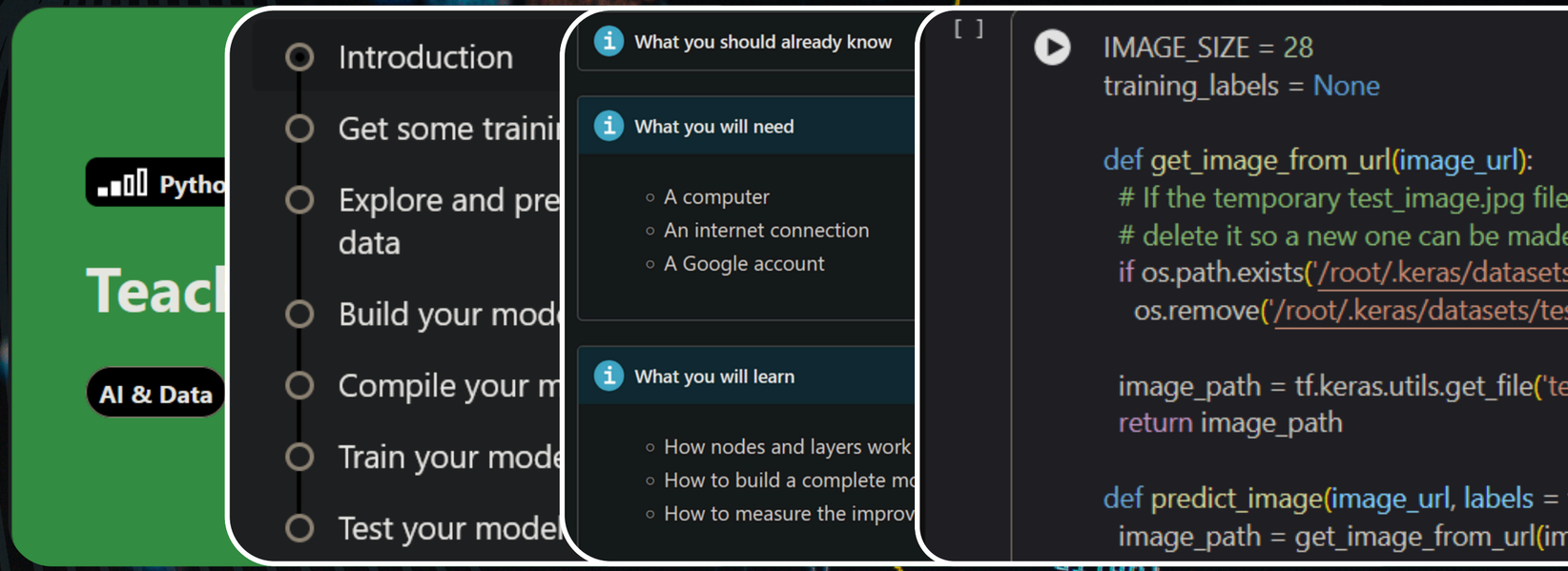
hands-on workshop

Open the project link below to start your Google Colab notebook.

https://projects.raspberrypi.org/en/projects/teach-a-computer-to-read/1

Steps:

- Load the Data.
- Build the Model Layers.
- Train for 10 Epochs.
- Test the model.



coming up next

Now that you have built a model, we will look at how these technologies apply to the real world.

Future Weeks:

- Real-World Showcases: How companies use what we just built.
- New Developments: Breaking news in AI, cutting-edge tools, and live demos.
- Student Projects: Opportunities for you to present your own findings.

