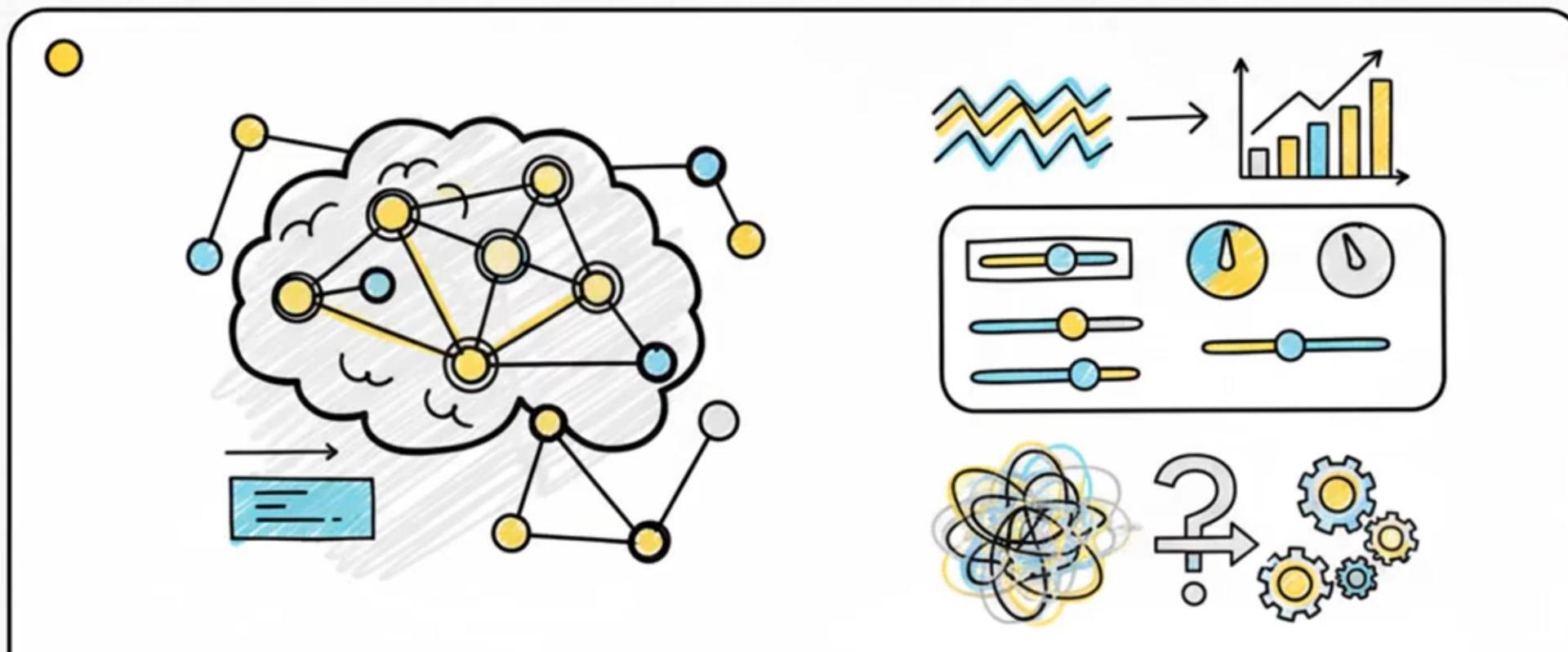
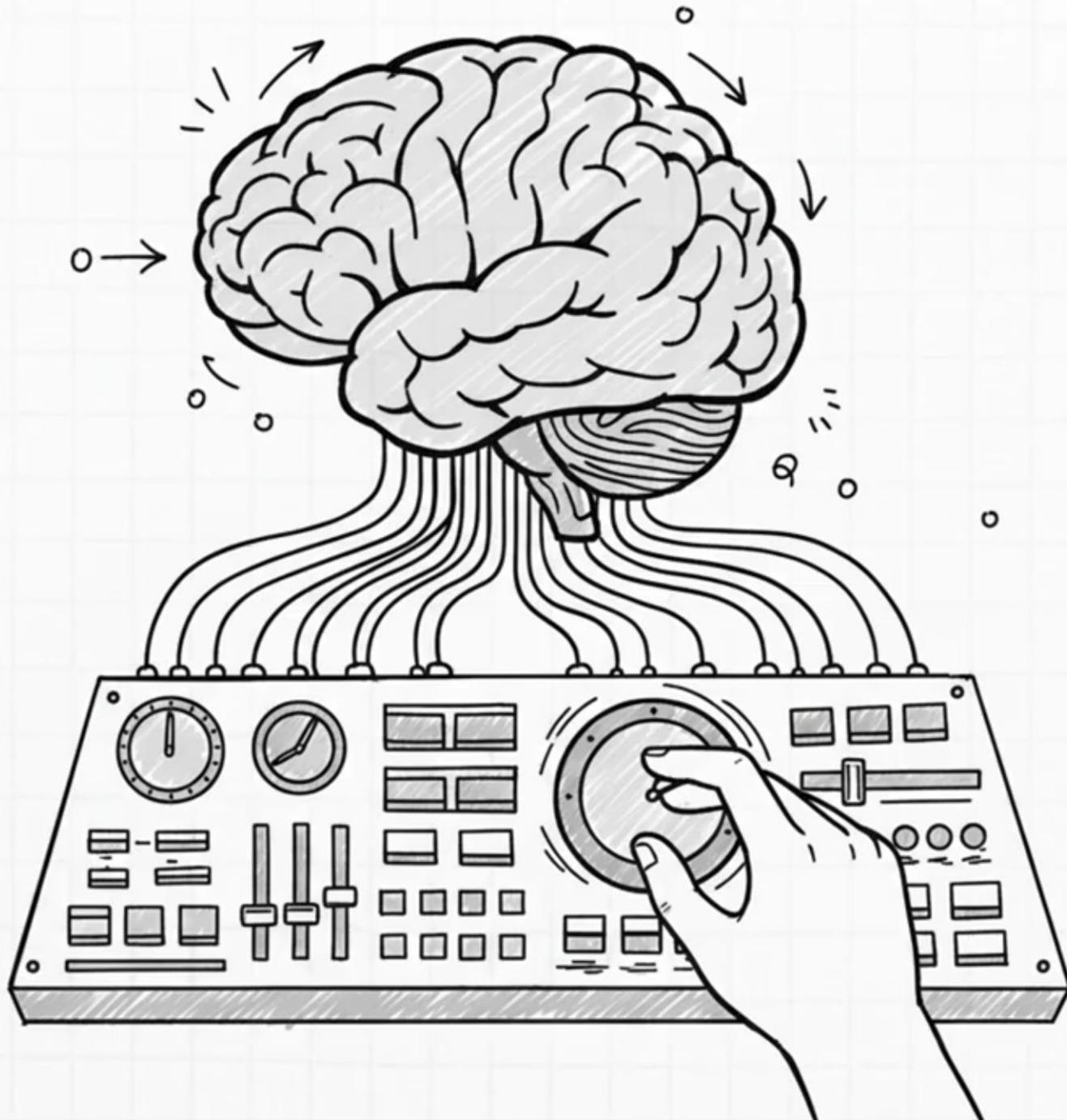
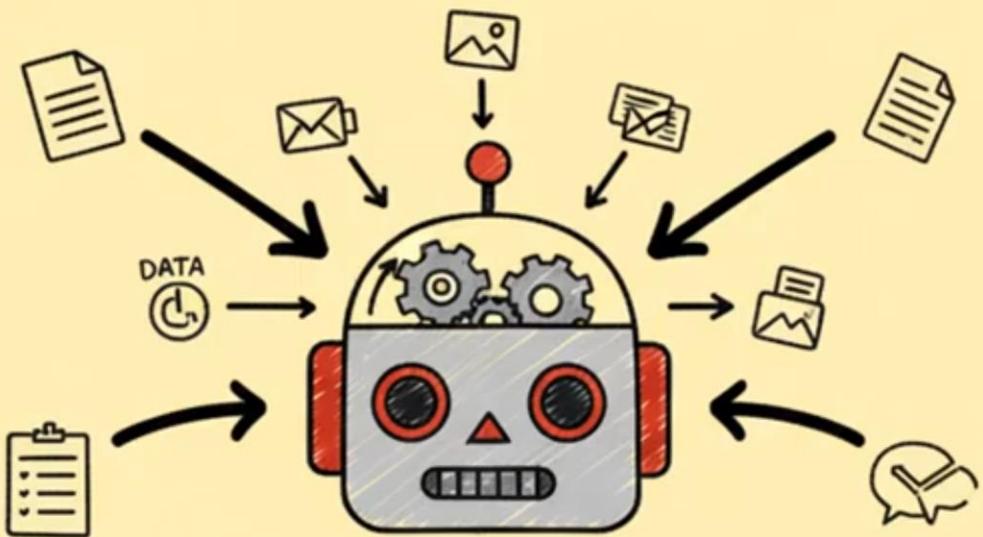


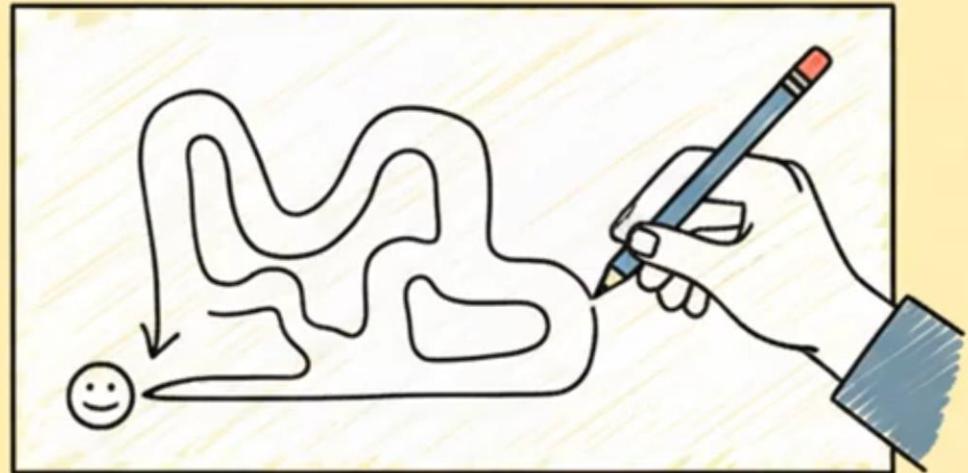
AI's Control Panel



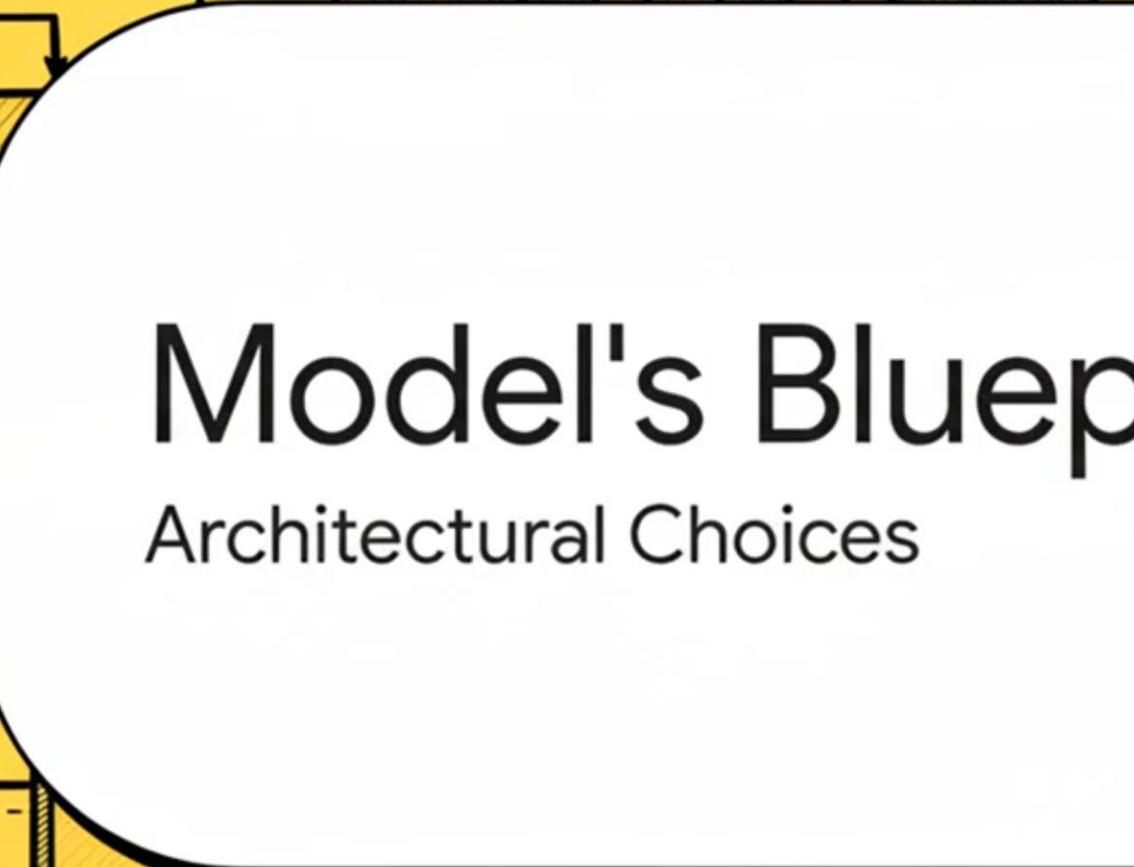




Parameters: What the model **learns** on its own from data, like internal weights.

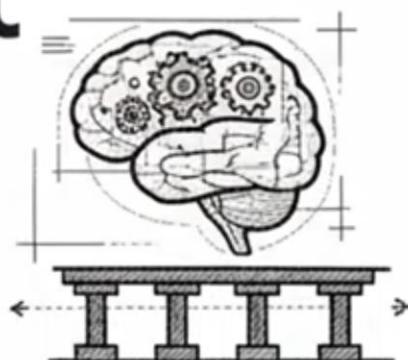


Hyperparameters: What you set **before** training to guide the learning process.



Model's Blueprint

Architectural Choices



Embedding Size

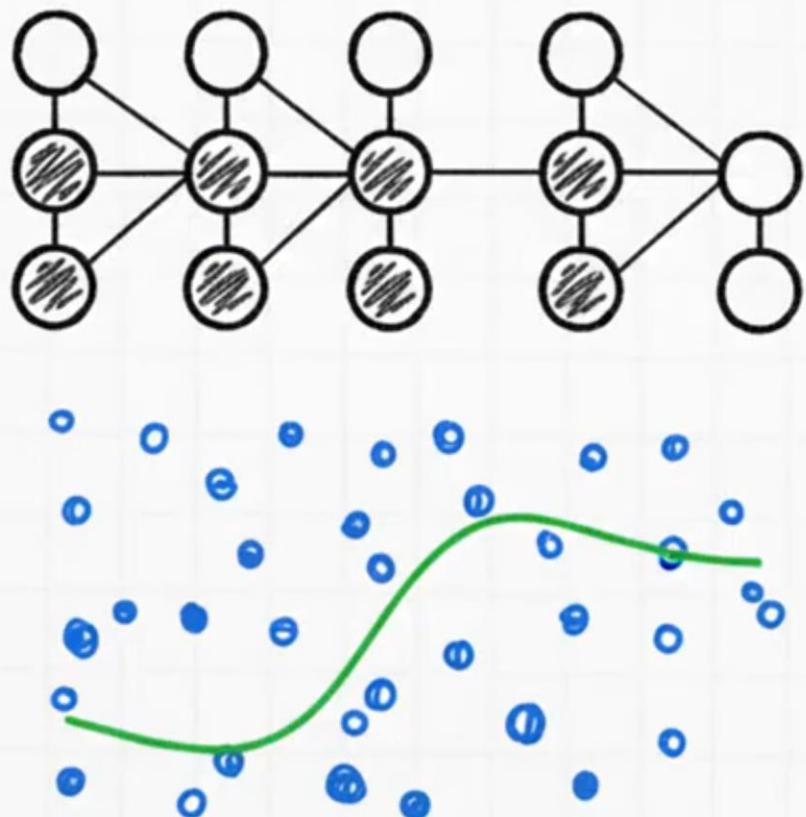
The length of the vector used to represent each discrete item, like a word, token, user, or product.

Project Type	Recommended Size
Tiny / Toy Project	32 or 64
Medium Problem	128 or 256
Large Language Model	768 to 4096+

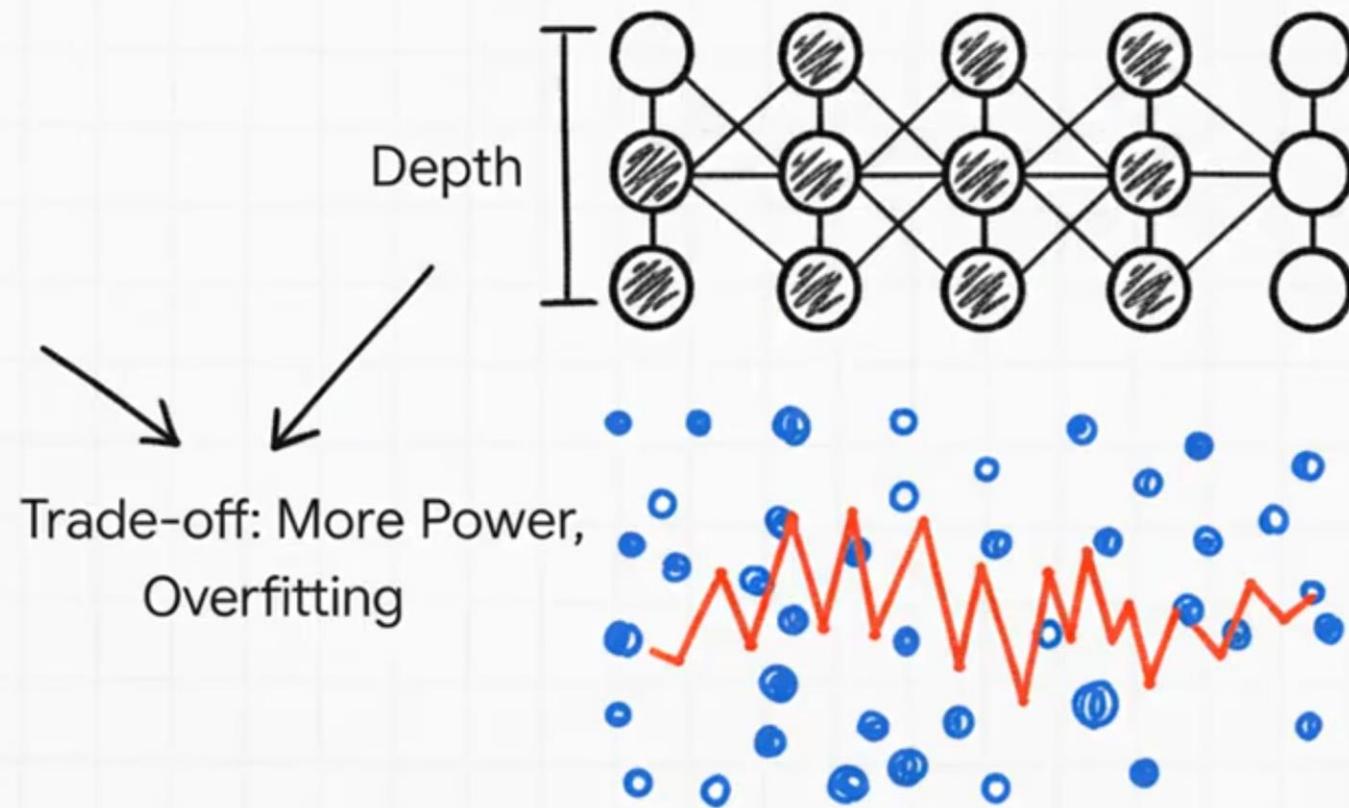
A decorative graphic surrounds the table. At the top left is a magnifying glass icon. Along the top edge are three horizontal wavy lines. On the left side, there's a yellow speech bubble containing a black checkmark. On the right side, there's a large spiral notebook icon with a yellow brain inside and two small circular icons with yellow swirls. Along the bottom edge are three horizontal wavy lines. In the bottom center is a bar chart icon with four bars of increasing height.

Layers & Hidden Units

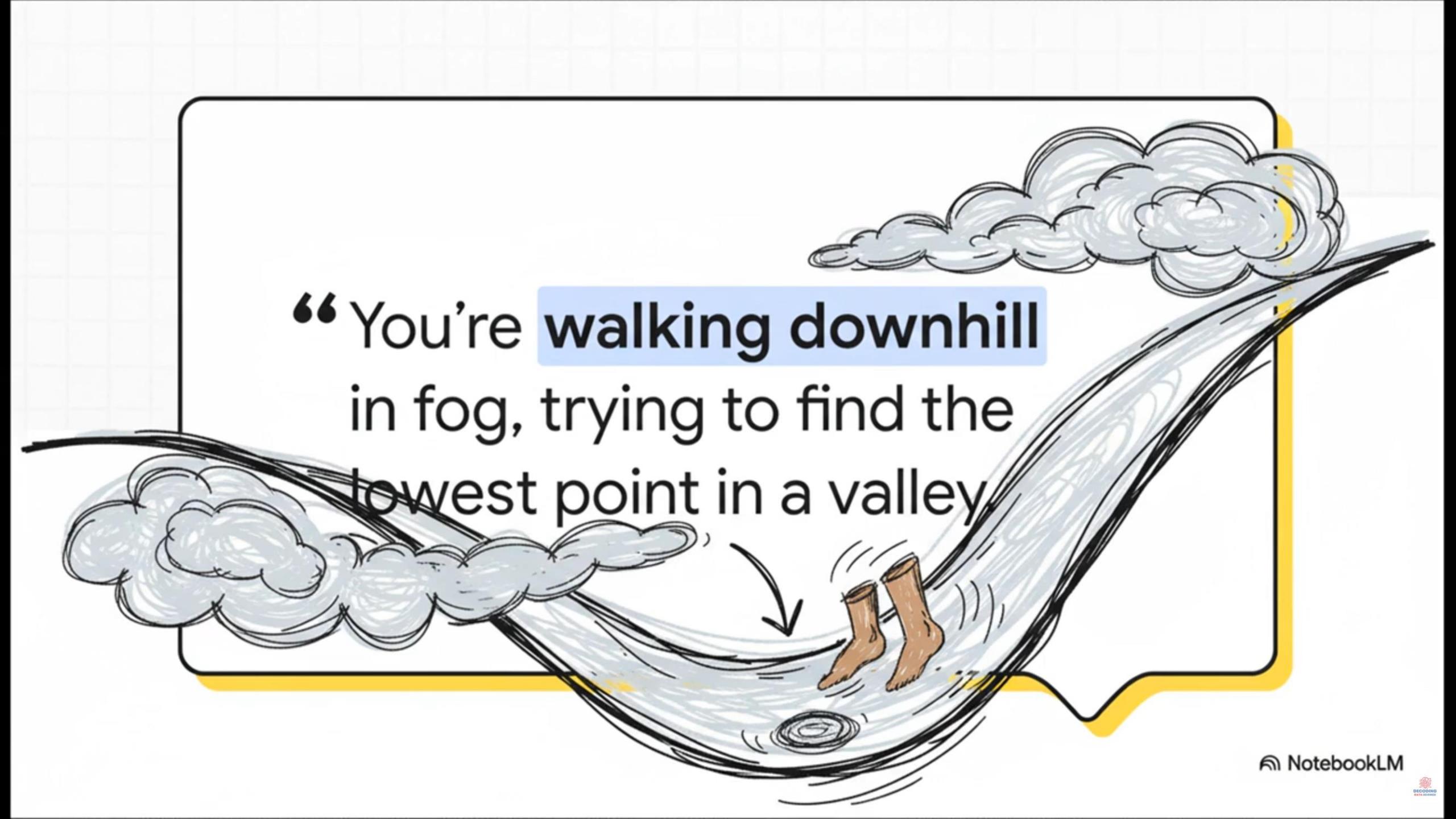
Simple Network



Complex Network



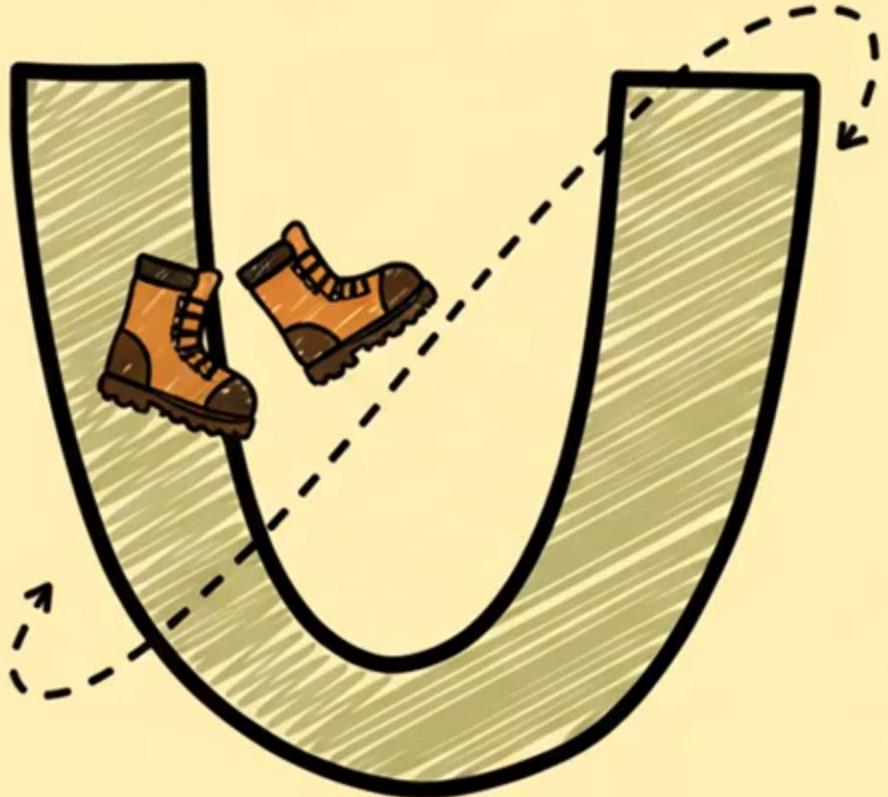
Depth
Trade-off: More Power,
Overfitting



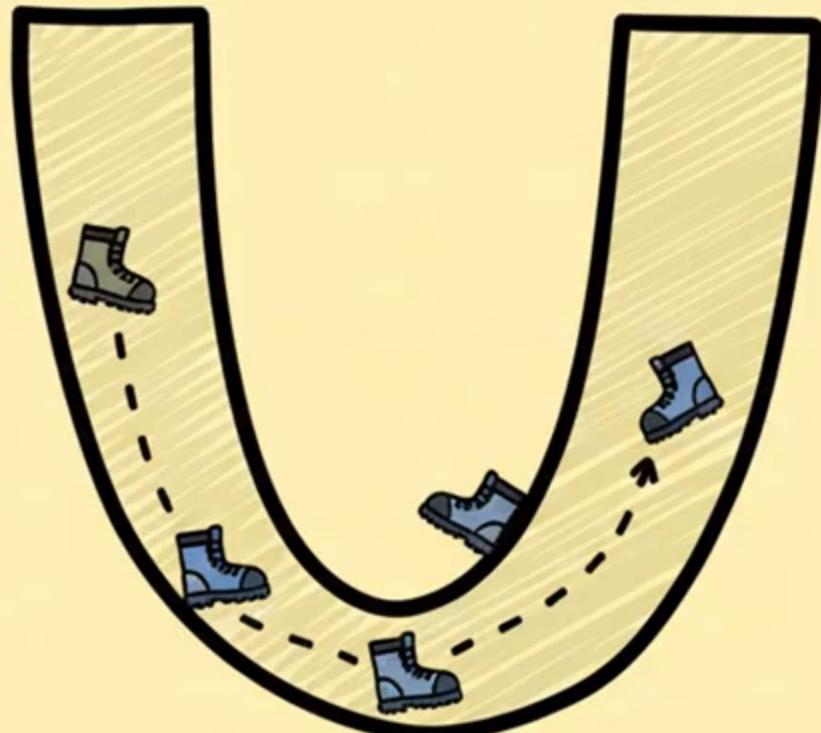
“You’re **walking downhill**
in fog, trying to find the
lowest point in a valley



Big Steps:



Tiny Steps:

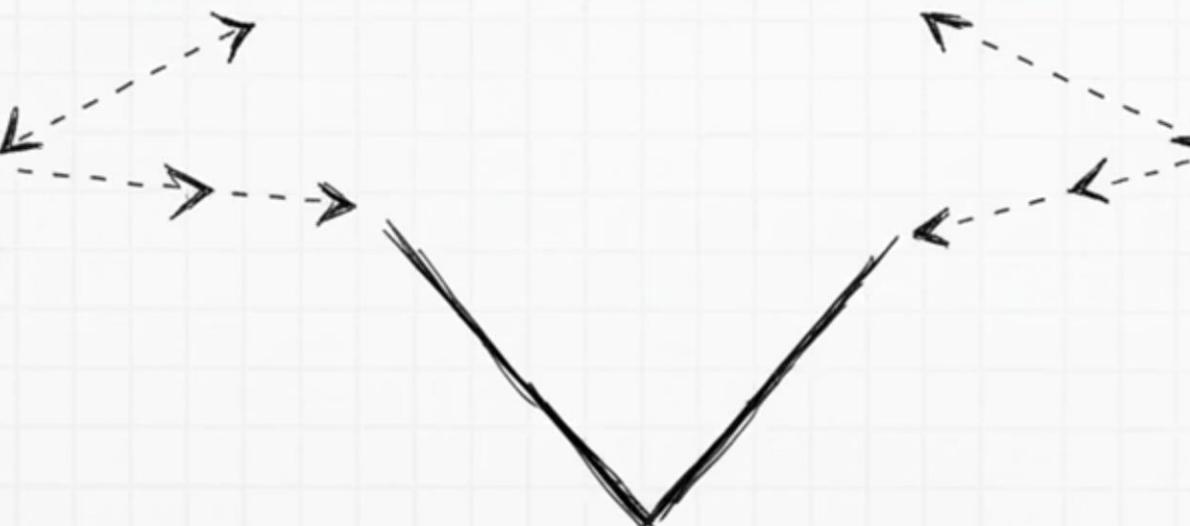


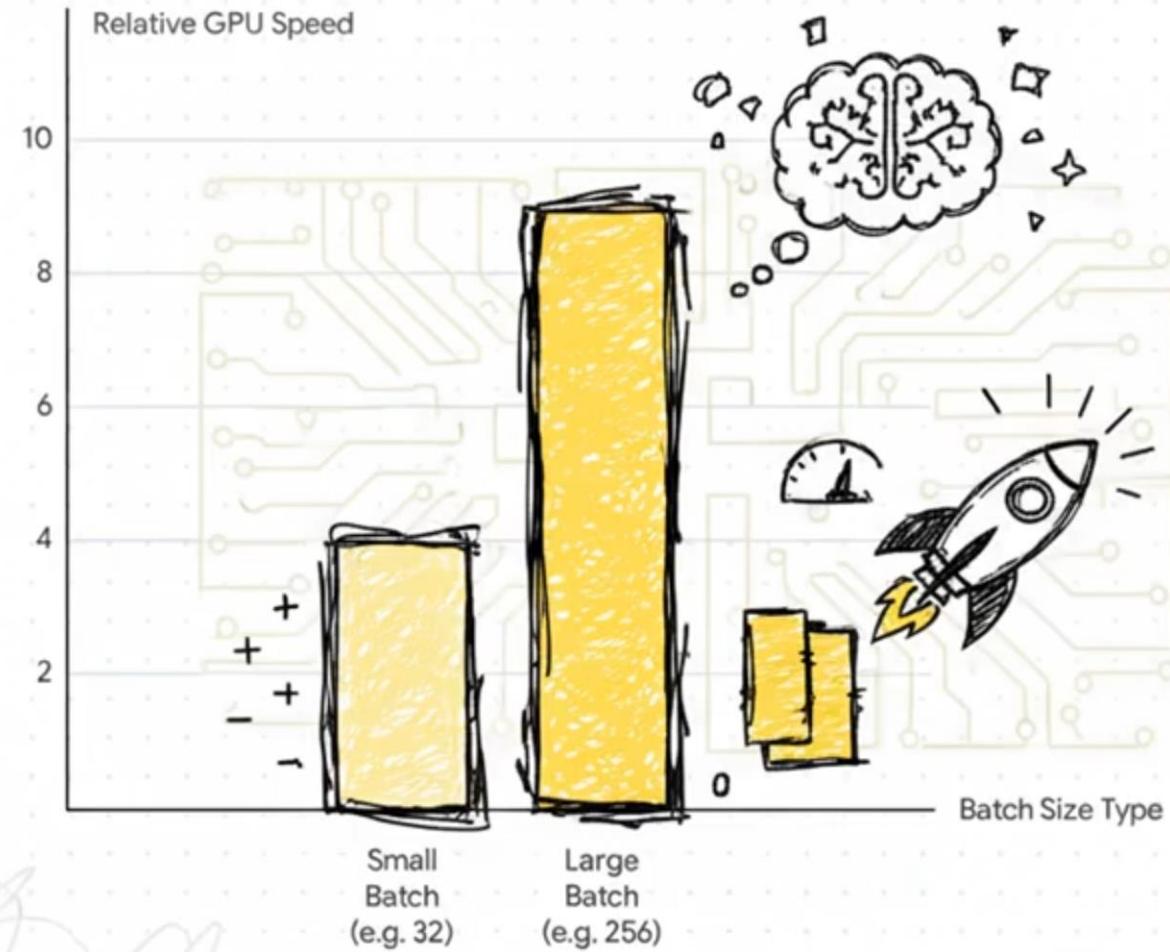


small, noisy
focus group

Batch Size

How many training examples you process before one weight update.



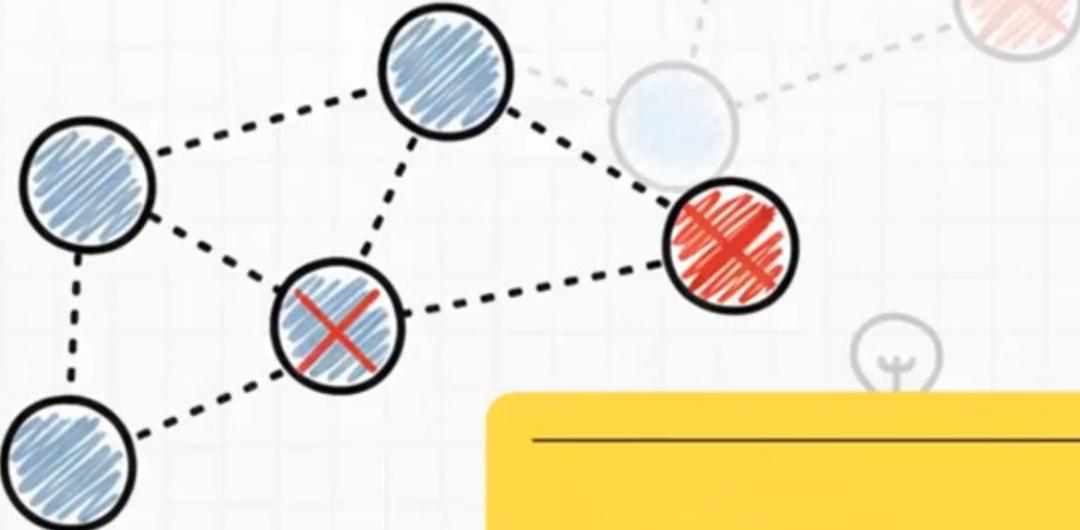


A key trade-off between
generalization and *training*
speed on a GPU.



Safety Guardrails

Preventing Overfitting



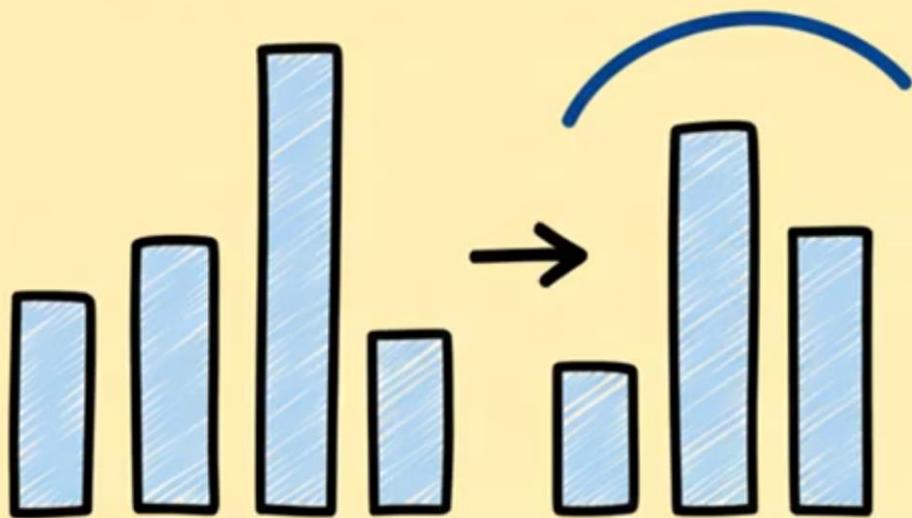
Dropout Rate

During training, randomly 'turn off' a fraction of neurons to force the network to learn robustly.

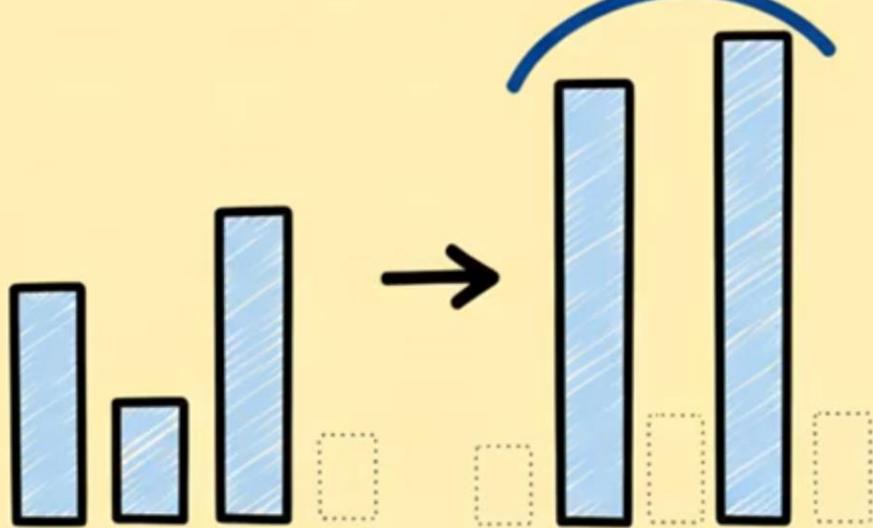




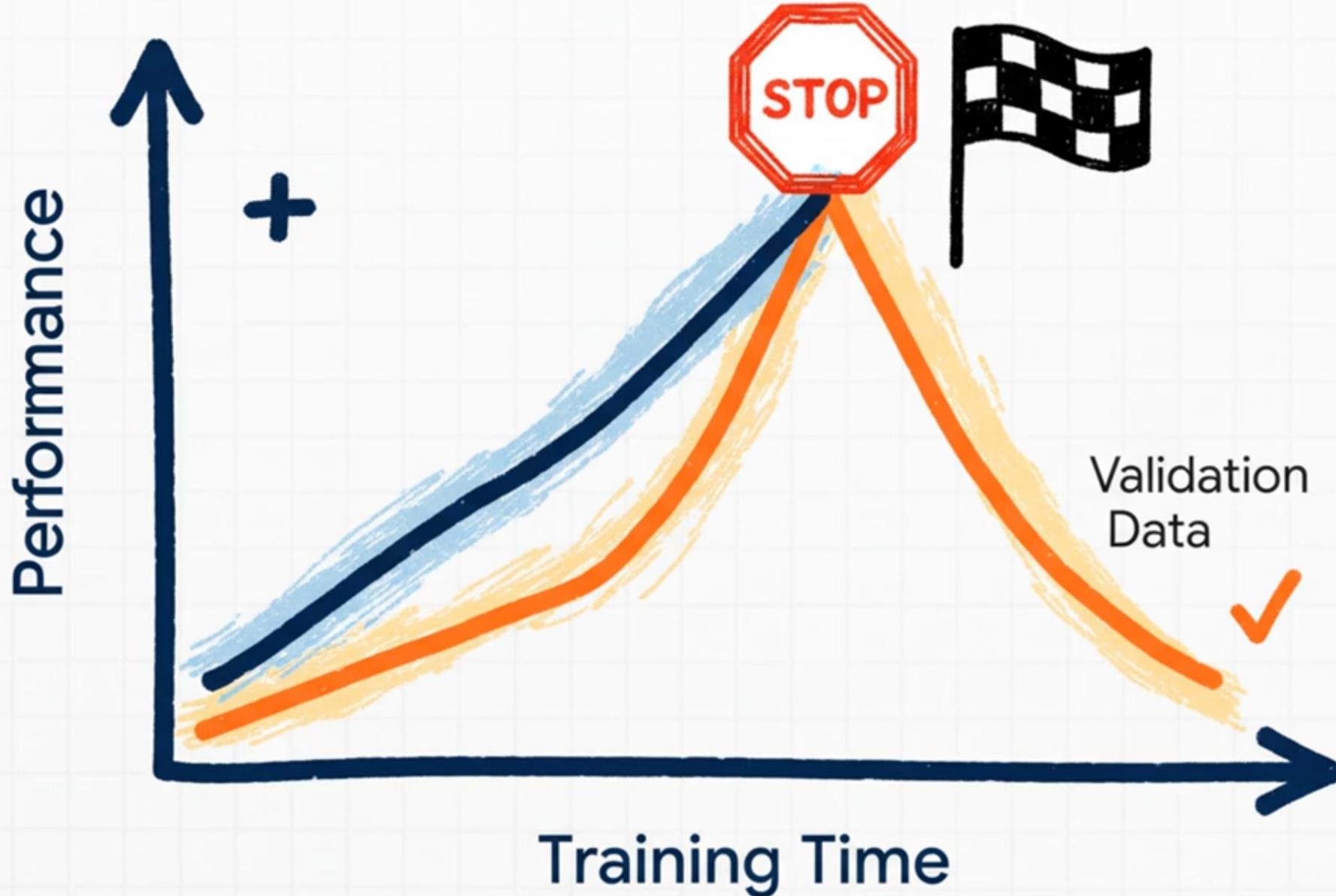
L2 (Weight Decay):



smoother model



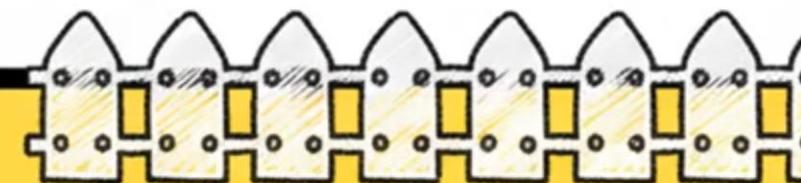
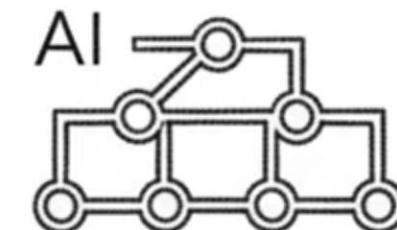
sparse model



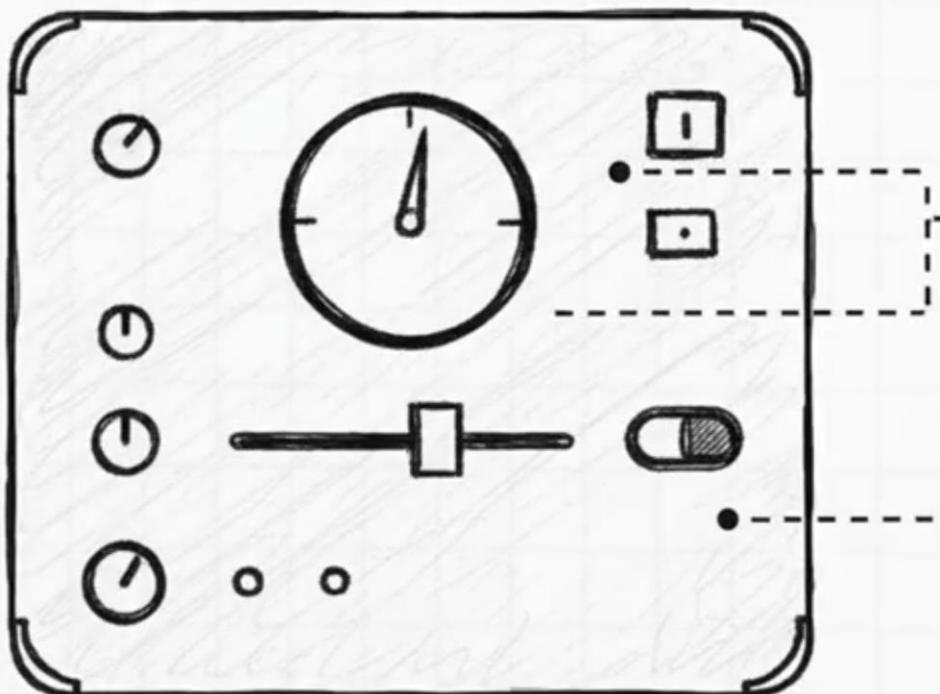
4

Tuning Strategy

From Theory to Practice



Hyperparameters in Code



- embedding_size = 128
- hidden_size = 256
- batch_size = 32
- learning_rate = 1e-3
- weight_decay = 1e-4
- dropout_rate = 0.3

Tuning Strategy

1. Start Simple

Begin with a basic architecture (2-4 layers).

2. Tune LR First

Find a rate where the loss goes down smoothly.

3. Adjust Batch Size

Use the largest your GPU can handle, then adjust if unstable.

6. Scale Up

Only now, try making your model bigger (more layers/units).

5. Add Regularization

If overfitting, add or increase dropout or weight decay.

4. Find Epochs

Use early stopping on a validation set to determine training length.



“Think of training as a **walk**. These dials control your step size, path noise, and how long you walk.





You have the **keys** to
the control panel. What
will you **build**?

