AlexNet — “The Big Dropout Net”

Rhyme:
“Two towers tall, ReLUs all;
Big filters start, five-by-five call.
Dropout guards when data’s small —
AlexNet wins it all.”

What this helps you recall

- Two-stream GPU split → “two towers tall”
- ReLU activation → “ReLUs all”
- First conv uses large filters (11×11, then 5×5) → “big filters… five-by-five call”
- Dropout introduced → “dropout guards”
- ImageNet 2012 winner → “wins it all”

ZFNet — “The Zoom-and-Fix Net”

Rhyme:
“Alex was wide, ZF zoomed inside;
Seven-by-seven to see the stride.
Deconv shows where features hide —
Fix the net and ride.”

✓ *What this helps you recall*

- Improved AlexNet → “Alex was wide… ZF zoomed inside”
- Changed 11×11 → 7×7 first filter → “seven-by-seven”
- Reduced stride for better feature capture → “see the stride”
- Introduced deconv visualizations → “where features hide”

GoogLeNet (Inception) — “The Multi-Path Net”

Rhyme:
“One net, many paths in sight;
One-three-five in fusion tight.
Bottlenecks make the model light —
GoogLeNet keeps depth just right.”

✓ *What this helps you recall*

- Inception module = many parallel paths → “many paths in sight”
- Uses 1×1, 3×3, 5×5 kernels together → “one-three-five in fusion tight”
- Uses 1×1 bottlenecks for dimensionality reduction → “bottlenecks… light”
- Very deep but efficient → “depth just right”

ResNet — “The Skip-Connection Net”

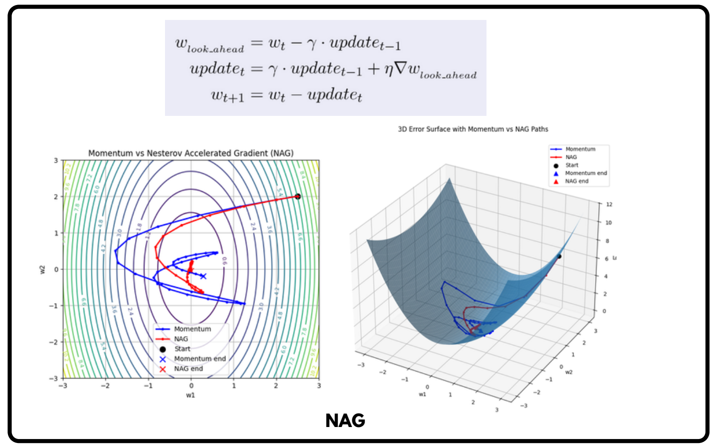
Rhyme:
“If learning fails, skip the trail;
Add it back so gradients sail.
Hundreds deep without a bail —
ResNet wins the trail.”

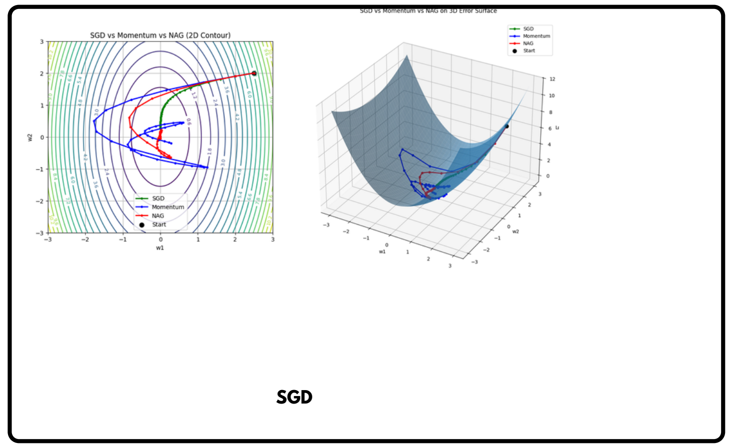
✓ *What this helps you recall*

- Skip connections / identity mapping → “skip the trail”
- Solves vanishing gradient → “gradients sail”
- Extremely deep models (50/101/152) → “hundreds deep”
- Breakthrough performance → “wins the trail”

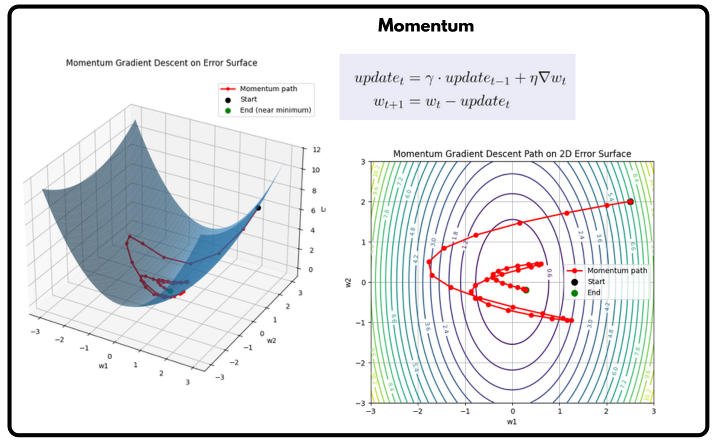
Gradient Descend

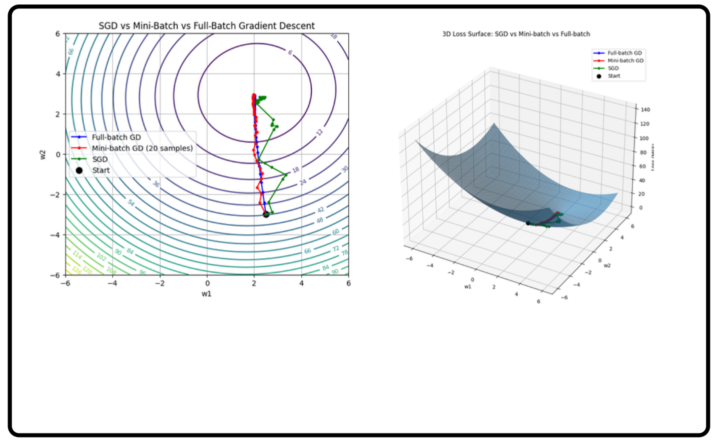
“Batch is slow but steady in sight,
SGD jumps left and right.
Mini-batch makes learning tight,
Momentum rolls with extra might,
Nesterov looks ahead for the right-step light.”





Gradient descend





- **Vanila Gradient Descend calculate gradient based on all sample points , then add all the calculated gradients and at last update the parameters like weight and biases using the summed gradient.**
- **The momentum and NAG variant of GD does the same only the update rule is different.**
- **In SGD, we calculate gradient on each sample point and update the parameters on each sample point. This does not lead to smooth convergence to optimal pint.**
- **The mini--batch version of SGD solved the problem by delaying the parameter update upto a predefined instance of gradient update**

Gradient $\longrightarrow$ $g = \frac{1}{N}\sum_{i=1}^{N} \nabla \ell_i$

	Vanilla GD	Momentum	NAG
Look Ahead	NA	NA	$\tilde{w} = w - \eta\beta v$
Velocity	NA	$v = \beta v + g$	$v = \beta v + \nabla \ell(\tilde{w})$
Update	$w = w - \eta g$	$w = w - \eta v$	$w = w - \eta v$