WACV' 18

OPTIMIZATION METHODS FOR DEEP LEARNING – THEORY AND PRACTICE

Sathya Ravi, Yunyang Xiong
Department of Computer Sciences
University of Wisconsin–Madison

13/03/2018

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

SCHEDULE

- Session I: 8:30 am to 9:30 am

  Focus: How to train machine learning models?

- Session II: 9:45 am to 10:45 am

  Focus: Why do these techniques "work"?

- Session III: 11 am to 11:45 am

  Focus: Practical examples

Material is made from papers/discussions/lecture notes/talks of Vikas Singh, Karl Rohe, Steve Wright, Rob Nowak, Ben Recht, Moritz Hardt, Dimitri Bertsekas, Kamalika Chaudhuri. Mistakes/incorrect statements are entirely due to me!

GRADIENT DESCENT (GD)

Solve $$\min_{W \in \mathbb{R}^n} L(W)$$

Do $$W_{t+1} \leftarrow W_t - \eta \nabla_W L(W)$$

until convergence

PRELIMINARIES

Taylor's theorem

$$L(W+d) = L(W) + \int_0^1 \nabla L(W+\gamma d)^T d \;\; d\gamma$$

$$L(W+d) = L(W) + \nabla L(W+\gamma d)^T d, \quad \text{for some} \quad \gamma \in (0,1)$$

PRELIMINARIES — II

Smoothness $\|\nabla L(U) - \nabla L(V)\| \leq \beta \|U - V\|$

$$\begin{aligned} L(V) - L(U) - \nabla L(U)^T (V - U) &= \int_0^1 [\nabla L(U + \gamma(V-U)) - \nabla L(U)]^T (V-U) d\gamma \\ &\leq \int_0^1 \|\nabla L(U + \gamma(V-U)) - \nabla L(U)\| \|V - U\| d\gamma \\ &\leq \int_0^1 \beta\gamma \|V-U\|^2 d\gamma \\ &= \frac{\beta}{2}\|V-U\|^2 \end{aligned}$$

We didn't need convexity at all!!

ANALYZE GD — I

$$L(W + \eta d) \leq L(W) + \eta \nabla L(W)^T d + \eta^2 \frac{\beta}{2} \|d\|^2$$

Recall the update rule: $W_{t+1} \leftarrow W_t - \eta \nabla_W L(W)$

$$\Rightarrow \quad L(W_{t+1}) \leq L(W_t) - \frac{1}{2\beta} \|\nabla L(W_t)\|^2$$

ANALYZE GD — II

$$\|\nabla L(W)\| \leq \sqrt{\frac{2\beta[L(W_0) - \bar{L}]}{T}}$$

Often $\bar{L} = 0$

LOCALLY GOOD

Let 0 be a fixed point for a local smooth map $\phi : U \to \mathbb{R}^n$ where U is a neighborhood of 0 Suppose $\mathbb{R}^n = E_s \oplus E_u$ where E_s is the span of the eigenvectors ≤ 1 of Jacobian at 0 and E_u the span of remaining. Then $\exists$ a disk tangent to E_s at 0 := local stable center manifold, and $\exists$ neighborhood B of 0 such that $\phi(disk) \cap B \subset disk$ and $\cap_{t=0}^{\infty} \phi^{-t}(B) \subset disk$.

Apply this to Gradient Descent to show that:

$$\mathbb{P}(\lim_t x_t = x_{\text{saddle}}) = 0$$

VARIANTS OF GD

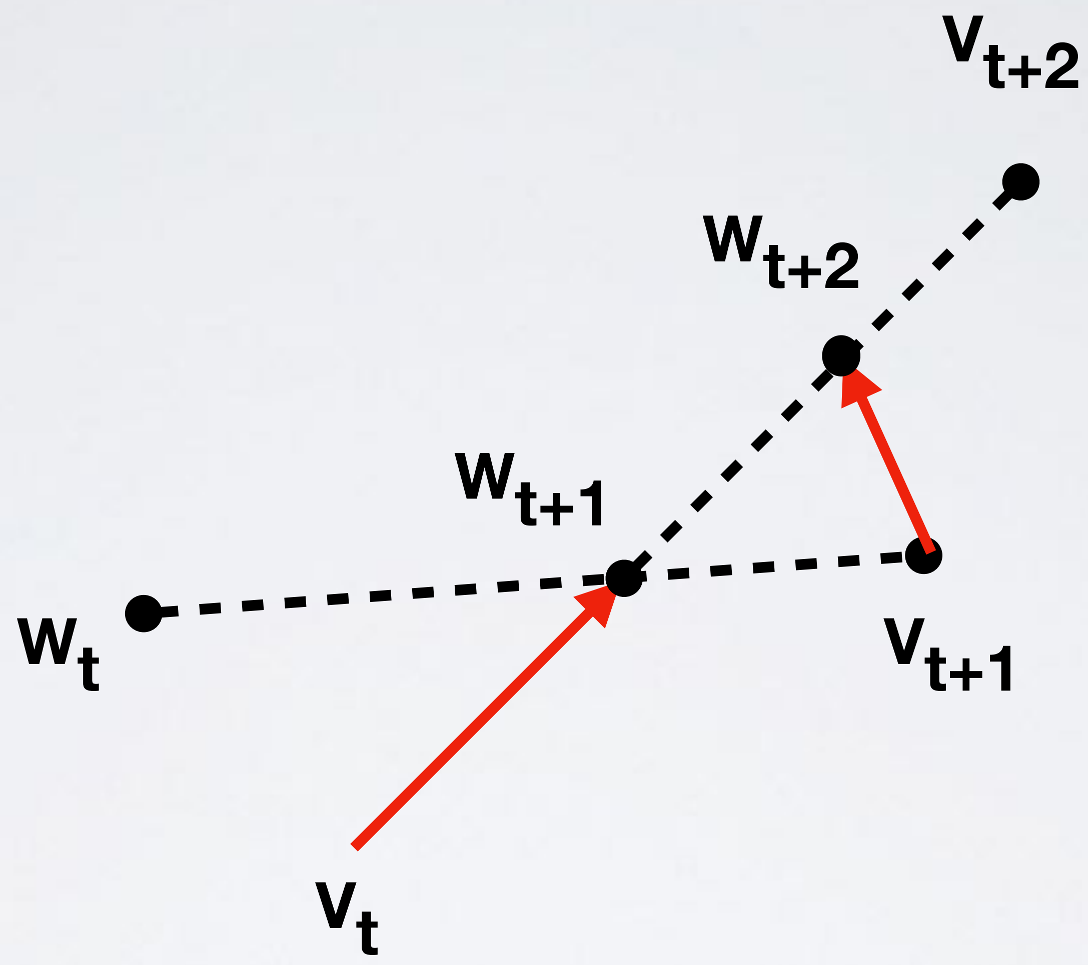
Different ways to choose η

- Exact line search
- Approximate line search
- Back tracking

One inequality to rule them all!

$$L(W_{t+1}) \leq L(W_t) - C\|\nabla L(W_t)\|^2$$

ACCELERATED GD



KEEPING UP WITH THE MOMENTUM

$$W_{t+1} = W_t - \eta \nabla L(W_t) + \alpha (W_t - W_{t-1})$$

Convergence is hard!

HOW FAST IS IT ANYWAY?

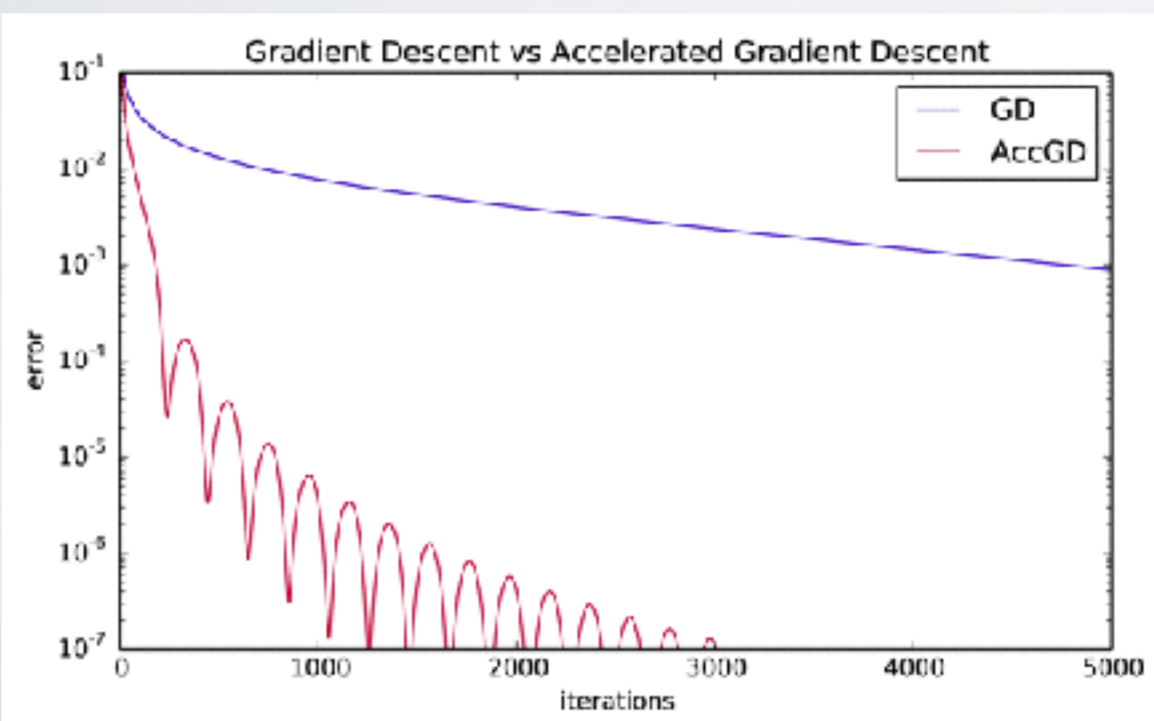
Method	Speed
GD	$O(1/eps^2)$
ACCELERATED GD	$O(1/eps^{7/4})$

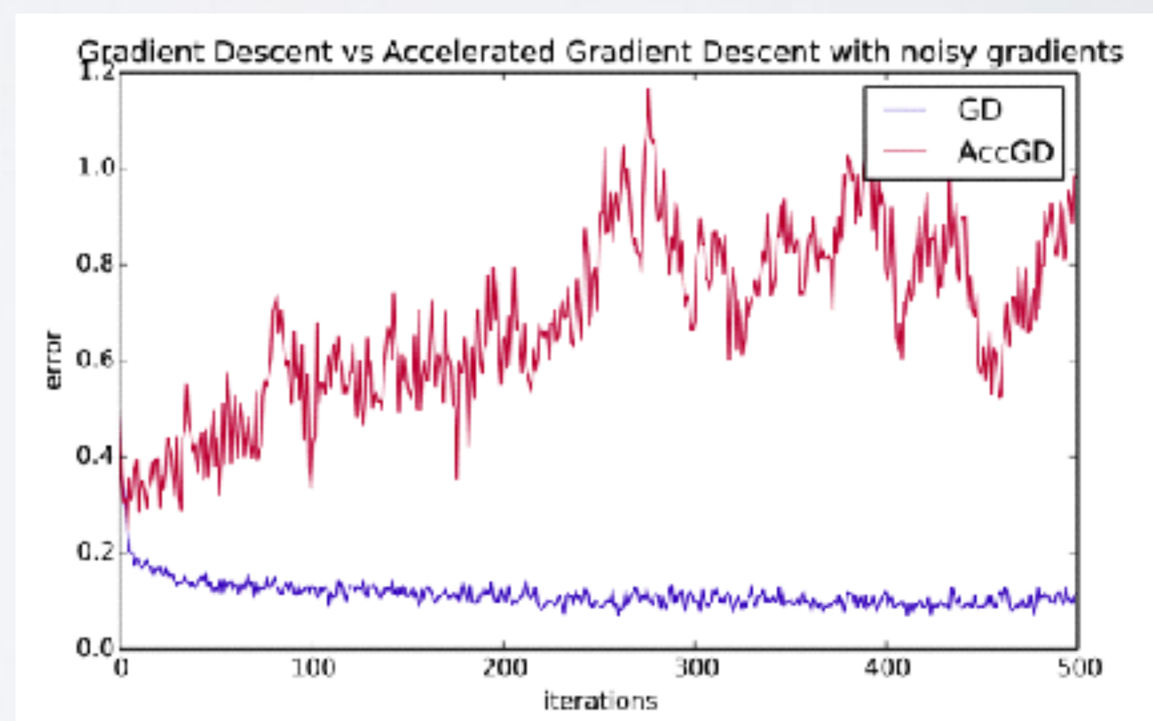
NOT THE END OF STORY!

Assume convexity and let's say we get a δ-approximate gradient at each time t.

Then Accelerated GD has: $L(W_t) - L^* \leq O(L/t^2) + O(t\delta)$

Then GD has: $L(W_t) - L^* \leq O(L/t) + O(\delta)$





YOU KEEP SAYING GRADIENT, BUT…

$$L(W) = \mathbb{E}_{\xi} f(W, \xi)$$

$$\xi = (x,y) \sim \mathscr{D}$$

How do I compute the gradient?

ENTER SGD

Compute an estimate of gradient

$$W_{t+1} = W_t - \eta_t \nabla \tilde{L}_t(W_t)$$

$$\mathbb{E}\left[\nabla \tilde{L}_t(W_t)\right] = \nabla L(W_t)$$

$$\mathbb{E}\left[\left\|\nabla \tilde{L}_t(W) - \nabla L(W)\right\|^2\right] \leq \sigma^2$$

ANALYZE SGD — I

$$L(W_{t+1}) \leq L(W_t) - \eta_t \nabla \tilde{L}_t(W_t)^T \nabla L(W_t) + \frac{\eta_t^2}{2} \nabla \tilde{L}_t(W_t)^T \nabla^2 L(W_t) \nabla \tilde{L}_t(W_t)$$

$$\Downarrow$$

$$\mathbb{E}[L(W_{t+1})|W_t] \leq L(W_t) - \eta_t \mathbb{E}[\nabla \tilde{L}_t(W_t)^T \nabla L(W_t)|W_t] + \frac{\eta_t^2 \beta}{2} \mathbb{E}[\|\nabla \tilde{L}_t(W_t)\|^2 |W_t]$$

$$\Downarrow$$

$$\eta_t < \frac{1}{\beta} \implies \mathbb{E}[L(W_{t+1})|W_t] \leq L(W_t) - \frac{\eta_t}{2} \|\nabla L(W_t)\|^2 + \frac{\eta_t^2 \sigma^2 \beta}{2}$$

ANALYZE SGD — II

$$\mathbb{E}[L(W_T)] \leq L(W_0) - \sum_{t=0}^{T-1} \frac{\eta_t}{2}[\|\nabla L(W_t)\|^2] + \sum_{t=0}^{T-1} \frac{\alpha_t^2 \sigma^2 \beta}{2}$$

$$\eta_t = \frac{\eta_0}{t+1} \implies \sum_{t=0}^{T-1} \frac{\eta_0}{2(t+1)}[\|\nabla L(W_t)\|^2] \leq L(W_0) - \mathbb{E}[L(W_T)] + \sum_{t=0}^{T-1} \frac{\eta_0^2 \sigma^2 \beta}{2(t+1)^2}$$

ANALYZE SGD — LAST PHEW!

$$Z_T = W_t \text{ with probability } \frac{1}{H_T(t+1)} \text{ where } H_t = \sum_{t=0}^{T-1} \frac{1}{t+1}$$

$$\mathbb{E}[\|\nabla L(Z_T)\|^2] = \sum_{t=0}^{T-1} \frac{1}{H_T(t+1)} \mathbb{E}[\|\nabla L(W_t)\|^2]$$

$$\lim_{T\to\infty} \mathbb{E}[\|\nabla L(Z_T)\|^2] = 0$$

WHAT DID WE MISS?

- Second Order Methods
- Stochastic Variance Reduced Methods
- SG — Langevin Dynamics
- Quantized Methods
- Constrained Optimization

QUESTIONS?

SEE YOU IN 15 MINUTES!

WACV' 18

OPTIMIZATION METHODS FOR DEEP LEARNING – THEORY AND PRACTICE II

Sathya Ravi, Yunyang Xiong
Department of Computer Sciences
University of Wisconsin–Madison

13/03/2018

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

RECAP

- What do we know so far?

  Computationally great

- Says nothing about learning!

  After all, that’s what we care about, isn’t it?

GENERALIZATION ERROR

$$\mathcal{R}(W) = \mathbb{E}_{(x,y)\sim\mathcal{D}} L(W;(x,y))$$

$$\mathcal{R}_S(W) = \frac{1}{n}\sum_{i=1}^{n} L(W;(x_i,y_i))$$

The one true theorem

$$\mathcal{R}(W) = \underbrace{\mathcal{R}_S(W)}_{\text{Train error}} + \underbrace{\mathcal{R}(W) - \mathcal{R}_S(W)}_{\Delta_S(W) := \text{Test error}}$$

LEARNING THEORY — 101

Occam's Razor: Simpler explanations should always be preferred

What do we mean by "simple"?

$$\mathfrak{R}_{n,D}(\mathcal{W}) = \mathbb{E}_{S\sim\mathcal{D}^{2n}}\left[\frac{1}{2n}\sup_{W\in\mathcal{W}}\left|\sum_{i=1}^{2n}\sigma_i L\left(W,(x_i,y_i)\right)\right|\right]$$

$\sigma_i = +1, -1$ with equal probability

WHY DO WE CARE?

$$\Delta_S(W) \lesssim 2\mathfrak{R}_{n,\mathcal{D}}$$

Proof (handwavy)

- Split S into S_1 and S_2
- For large enough m, $L_{S_2}(W) \cong L_D(W)$ and thus $L_D(W)$ - $L_{S_1}(W) \cong L_{S_2}(W)$-$L_{S_1}(W)$
- S_2 is like the training set and S_1 is the test set

Since S1 and S2 were randomly picked

$$\mathbb{E}_{S\sim\mathcal{D}^{2m}}\left[\mathbb{E}_{z\sim S_2}[L(W,z)]-\mathbb{E}_{z\sim S_1}[L(W,z)]\right] \leq \mathbb{E}_{S\sim\mathcal{D}^{2m}}\left[\frac{1}{2m}\left|\sum_i \sigma_i L(W,z_i)\right|\right]$$

$$\leq \sup_{W\in\mathcal{W}} \mathbb{E}_{S\sim\mathcal{D}^{2m}}\left[\frac{1}{2m}\left|\sum_i \sigma_i L(W,z_i)\right|\right]$$

$$\Downarrow$$

$$\Delta_S(W) \leq \sup_{W\in\mathcal{W}} \mathbb{E}_{S\sim\mathcal{D}^{2m}}\left[\frac{1}{2m}\left|\sum_i \sigma_i L(W,z_i)\right|\right]$$

$$\leq \mathbb{E}_{S\sim\mathcal{D}^{2m}} \sup_{W\in\mathcal{W}}\left[\frac{1}{2m}\left|\sum_i \sigma_i L(W,z_i)\right|\right] = 2\mathfrak{R}_{n,\mathcal{D}}(\mathcal{W})$$

EXAMPLES & SYNOPSIS

For linear classifiers

$$\mathcal{W} = \{W : \|W\|_2 \leq 1\} \implies \mathfrak{R}(\mathcal{W}) = O\left(\frac{\max_i \|x_i\|_2}{\sqrt{n}}\right)$$

$$\mathcal{W} = \{W : \|W\|_1 \leq 1\} \implies \mathfrak{R}(\mathcal{W}) = O\left(\frac{\max_i \|x_i\|_\infty \sqrt{\log d}}{\sqrt{n}}\right)$$

Summary

Low $\mathfrak{R}(\mathcal{W})$ is good!

BACK TO SGD

- Radamacher Complexity is algorithm and data agnostic and depends only on the richness/complexity of the hypothesis class/space $\mathscr{W}$. It is often referred to as "uniform convergence" since it works for any W in $\mathscr{W}$.
- Doesn't give us too much intuition about why the methods we use work well in practice
- So we need a different approach…

SGD — AN ÜBER ALGORITHM

Any model trained by SGD within a reasonable number of steps has vanishing generalization error

STABILITY ↠ GENERALIZATION

Small perturbations in the data don't change training loss much

A randomized algorithm A is $\epsilon -$ uniformly stable if for all datasets $S, S' \in \mathcal{D}^n$ such that S, S' differ in at most one example, we have,

$$\sup_z \mathbb{E}_A \left[L(A(S), z) - L(A(S'), z) \right] \leq \epsilon$$

STABILITY ↠GENERALIZATION II

Redefining generalization error

$$\epsilon_{\text{gen}} = \mathbb{E}_{S,A}\left[\mathcal{R}_S[A(S)] - \mathcal{R}[A(S)]\right]$$

Theorem

Let A be ϵ-uniformly stable. Then $\epsilon_{\text{gen}} \leq \epsilon$

LET'S PROVE IT!

- S, S' be two samples. S(i) be S except for the i-th data point where it is replaced from S'

$$
\begin{aligned}
\mathbb{E}_S\mathbb{E}_A[R_S[A(S)]] &= \mathbb{E}_S\mathbb{E}_A\left[\frac{1}{n}\sum_{i=1}^{n} L(A(S), z_i)\right] \\
&= \mathbb{E}_S\mathbb{E}_S'\mathbb{E}_A\left[\frac{1}{n}\sum_{i=1}^{n} L(A(S^i), z_i')\right] \\
&= \mathbb{E}_S\mathbb{E}_S'\mathbb{E}_A\left[\frac{1}{n}\sum_{i=1}^{n} L(A(S), z_i')\right] + \delta \\
&= \mathbb{E}_S\mathbb{E}_A[R[A(S)]] + \delta \\
&\leq \mathbb{E}_S\mathbb{E}_A[R[A(S)]] + \epsilon
\end{aligned}
$$

WHAT ABOUT SGD?

$$\epsilon_{\mathrm{stab}}^{\mathrm{SGD}} \lessapprox \frac{T^{1-\frac{1}{\beta+1}}}{n}$$

T = O(n) is good

PROOF IDEA

- Analyze the behavior of SGD for two datasets that differ by one example
- Use a Stopping time analysis
- SGD has a longer "burn-in period": where δ_t doesn't grow too much
- When δ_t does grow, η_t has decayed

Can easily handle other stability inducing operations
Weight Decay, Clipping etc..

Amenable to convex constraints too!

EXTENSIONS

- High probability bounds
- Uniform Hypothesis Stability
- Data dependent bounds using information theory

THINGS WE MISSED

- Uniform convergence of Deep Networks
- PAC-Bayes Based Approaches
- Differential Privacy
- Adversarial Training
- Generative Adversarial Networks

QUESTIONS?
SEE YOU IN 15 MINUTES!

WACV' 18

OPTIMIZATION METHODS FOR DEEP LEARNING – THEORY AND PRACTICE III

Sathya Ravi, Yunyang Xiong
Department of Computer Sciences
University of Wisconsin–Madison

13/03/2018

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

LET'S BE PRACTICAL

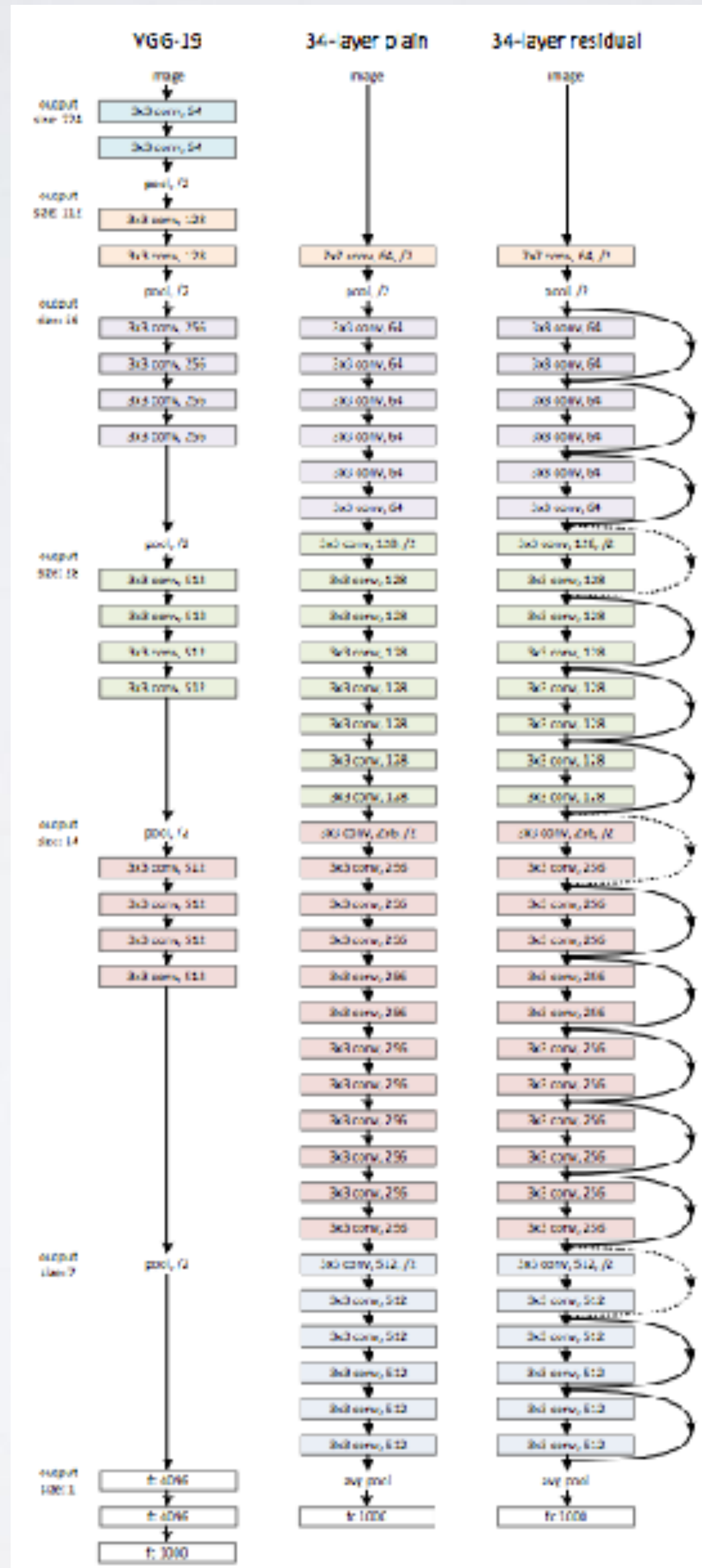
In theory, there is no difference between theory and practice. But, in practice, there is

GETTING DOWN TO BRASS TACKS

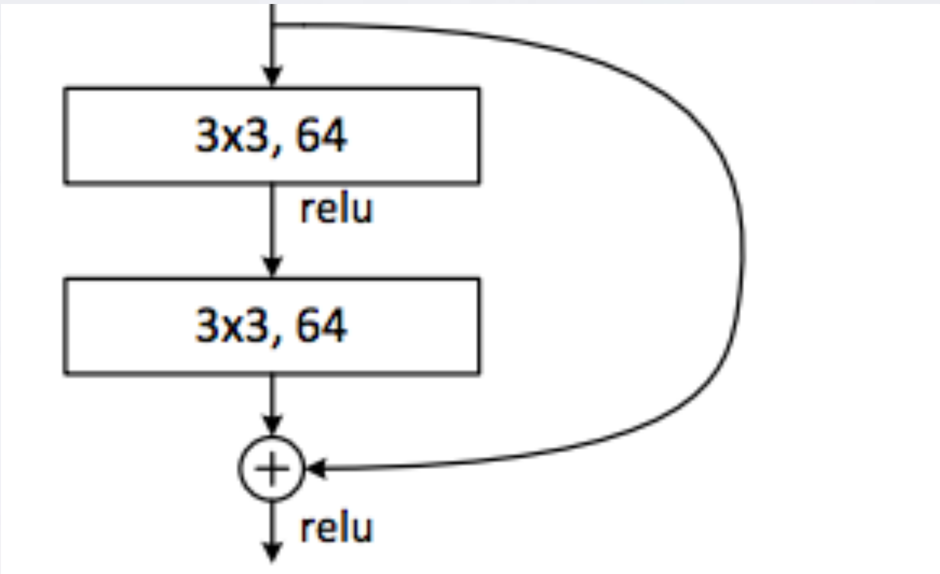
- **Choose framework**
- **Choose algorithm**
- **Run**

We will see THREE examples!

DEEP RESIDUAL NETWORKS

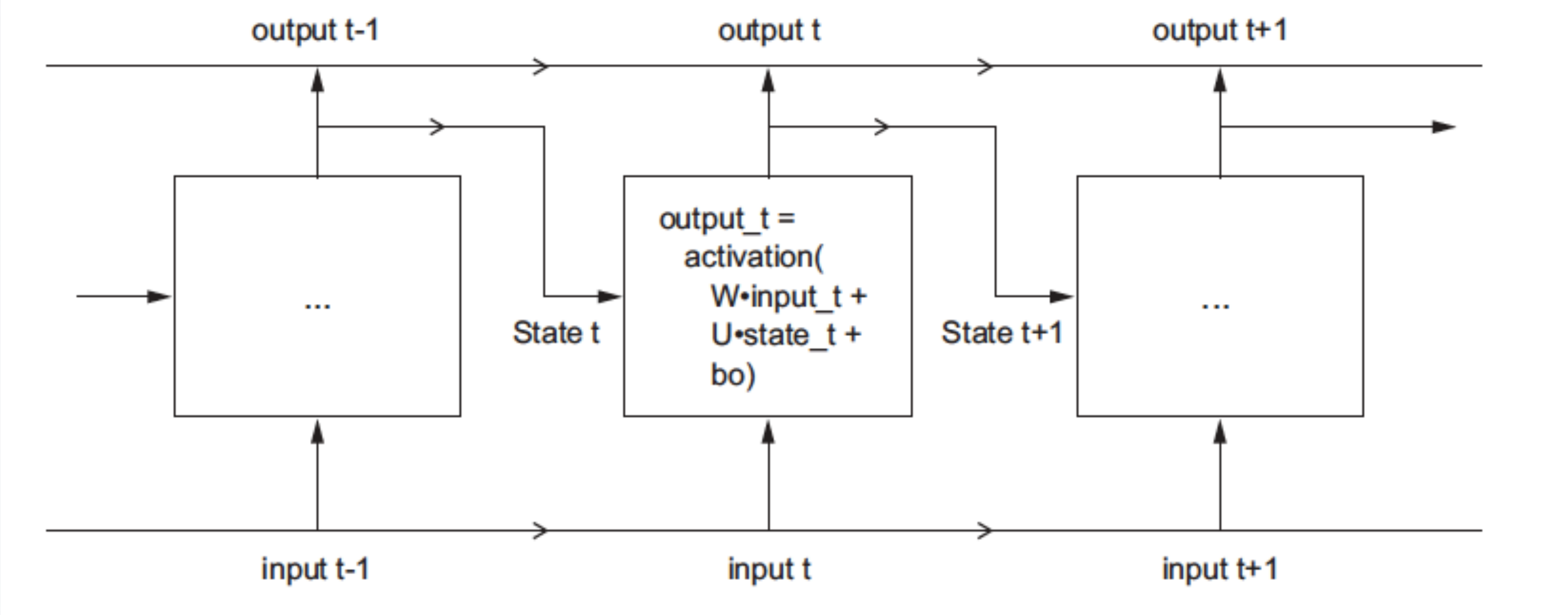


RESNET LAYERS

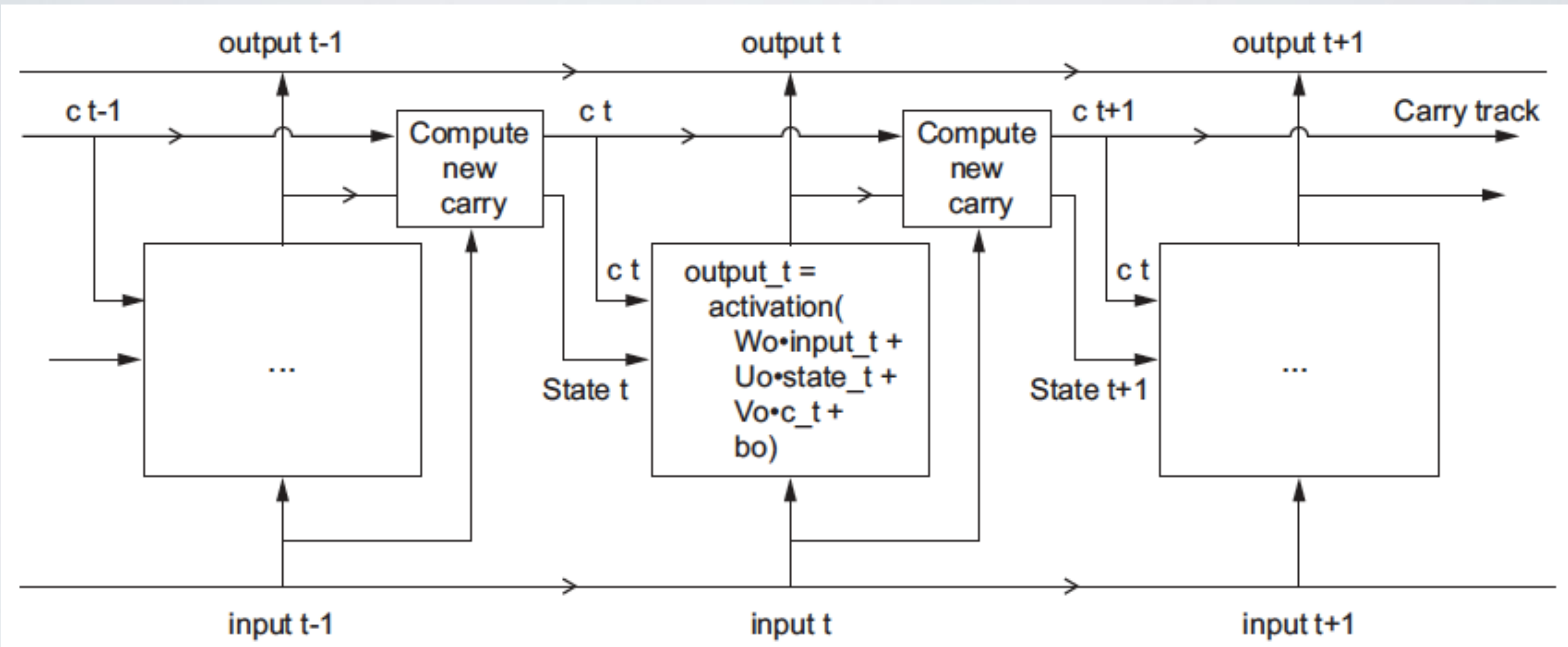


DEMO

RNN


output t-1
output t
output t+1
...
State t
output_t =
activation(
W•input_t +
U•state_t +
bo)
State t+1
...
input t-1
input t
input t+1

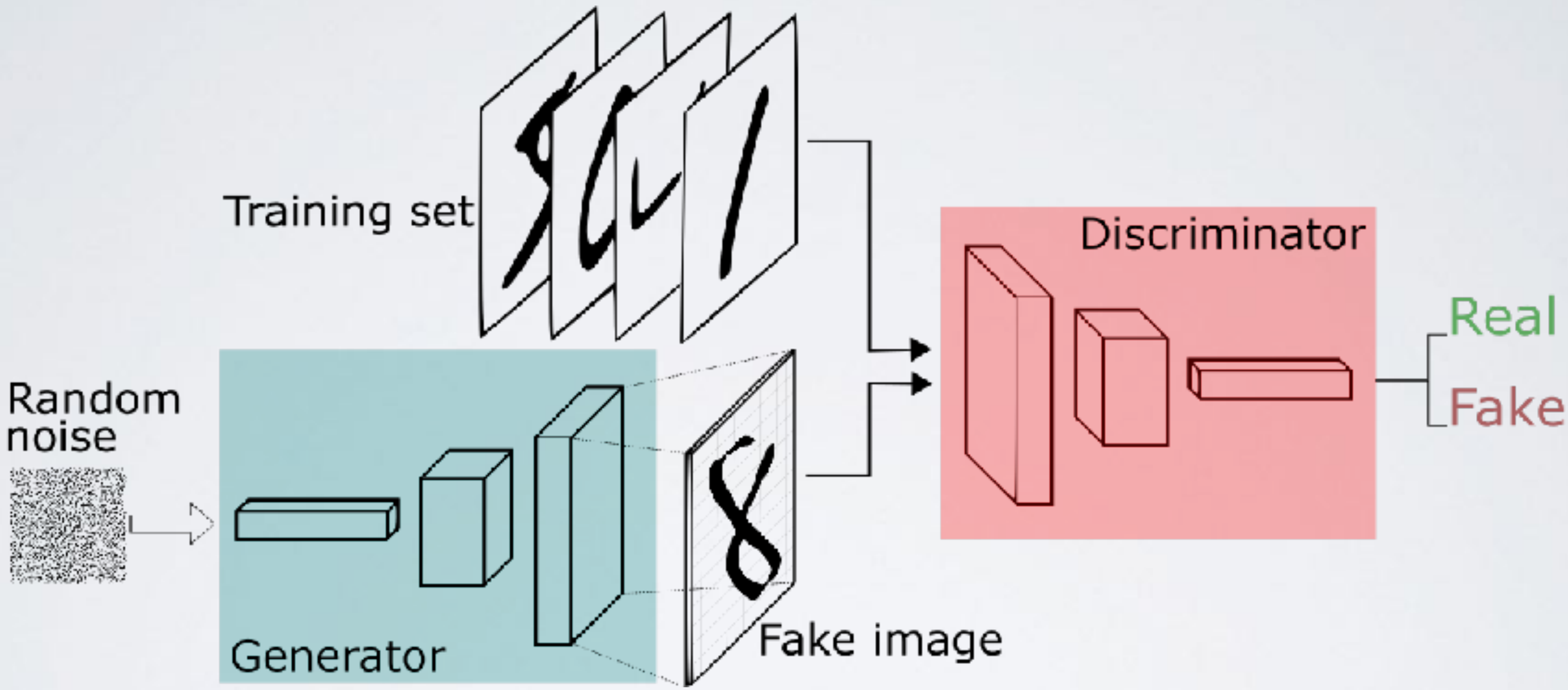
LSTM



output t-1
output t
output t+1
c t-1
Compute
new
carry
c t
c t+1
Carry track
c t
c t
...
output_t =
activation(
Wo•input_t +
Uo•state_t +
Vo•c_t +
bo)
State t
State t+1
...
input t-1
input t
input t+1

DEMO

GENERATIVE ADVERSARIAL NETWORKS (GAN)



GAN MATH

$$\min_G \max_D \mathbb{E}_{x \sim \mathcal{D}_{\text{real}}}[f(D(x))] + \mathbb{E}_h[f(1 - D(G(h)))]$$

DEMO

QUESTIONS?