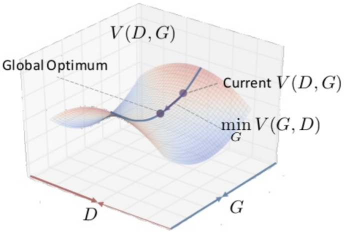
Deep Learning

Challenges for GAN Optimization



Learning goals

- (no) convergence to fix point
- problems of adversarial setting

ADVERSARIAL TRAINING

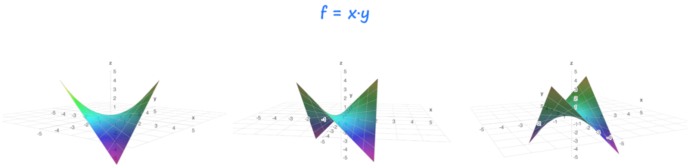
Deep Learning models (in general) involve a single player!

- The player tries to maximize its reward (minimize its loss).
- Use SGD (with backprob) to find the optimal parameters.
- SGD has convergence guarantees (under certain conditions).
- However, with non-convexity, we might convert to local minima!

GAN instead involve two players

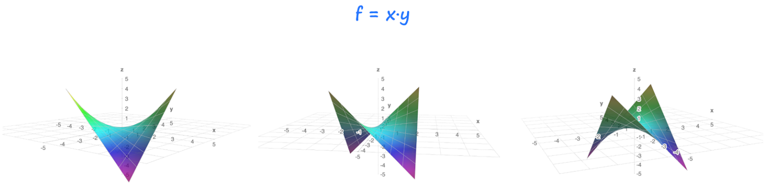
- Discriminator is trying to maximize its reward.
- Generator is trying to minimize discriminator's reward.
- SGD was not designed to find the Nash equilibrium of a game!
- Therefore, we might not converge to the Nash equilibrium at all!

ADVERSARIAL TRAINING -EXAMPLE



- Consider the function $f(x, y) = xy$, where x and y are both scalars.
- Player A can control x and Player B can control y.
- The loss:
  - Player A: $L_A(x, y) = xy$
  - Player B: $L_B(x, y) = -xy$
- This can be rewritten as $L(x, y) = \min_x \max_y xy$
- What we have here is a simple zero-sum game with its characteristic minimax loss.

POSSIBLE BEHAVIOUR #1: CONVERGENCE

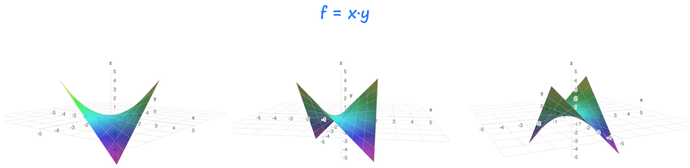


- The partial derivatives of the losses are:

$$\frac{\partial L_A}{\partial x} = y \, , \, \frac{\partial L_B}{\partial y} = -x$$

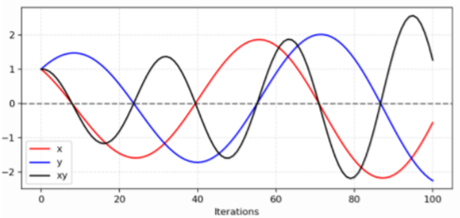
- In adversarial training, both players perform gradient descent on their respective losses.
- We update x with $x - \alpha \cdot y$ and y with $y + \alpha \cdot x$ simultaneously in one iteration, where α is the learning rate.

POSSIBLE BEHAVIOUR #1: CONVERGENCE



- In order for simultaneous gradient descent to converge to a fixed point, both gradients have to be simultaneously 0.
- They are both (simultaneously) zero only for the point (0,0).
- This is a saddle point of the function $f(x, y) = xy$.
  - The fixed point for a minimax game is typically a saddle point.
  - Such a fixed point is an example of a Nash equilibrium.
- In adversarial training, convergence to a fixed point is **not** guaranteed.

POSSIBLE BEHAVIOUR #2: CHAOTIC BEHAVIOUR



Credit: Lilian Weng

Figure: A simulation of our example for updating x to minimize xy and updating y to minimize -xy. The learning rate α = 0.1. With more iterations, the oscillation grows more and more unstable.

- Once x and y have different signs, every following gradient update causes huge oscillation and the instability gets worse in time, as shown in the figure.

POSSIBLE BEHAVIOUR #3: CYCLES

Credit: Goodfellow

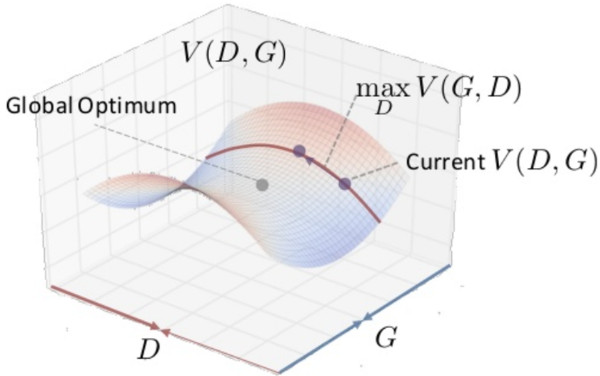
Figure: Simultaneous gradient descent with an infinitesimal step size can result in a circular orbit in the parameter space.

- A discrete example: A never-ending game of Rock-Paper-Scissors where player A chooses 'Rock' $\rightarrow$ player B chooses 'Paper' $\rightarrow$ A chooses 'Scissors' $\rightarrow$ B chooses 'Rock' $\rightarrow$ ...
- **Takeaway:** Adversarial training is highly unpredictable. It can get stuck in cycles or become chaotic.

NON-STATIONARY LOSS SURFACE

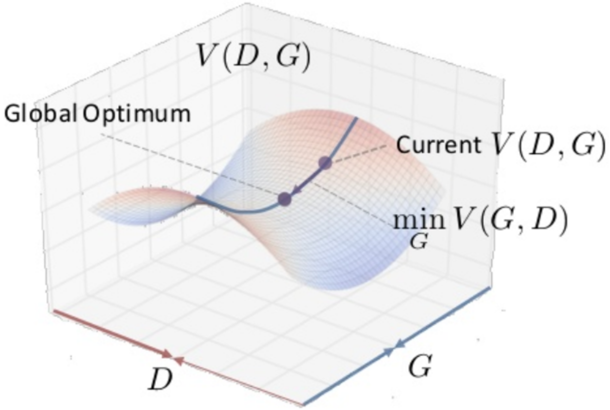
- From the perspective of one of the players, the loss surface changes every time the other player makes a move.
- This is in stark contrast to (full batch) gradient descent where the loss surface is stationary no matter how many iterations of gradient descent are performed.

ILLUSTRATION OF CONVERGENCE



Credit: Mark Chang

ILLUSTRATION OF CONVERGENCE: FINAL STEP



Credit: Mark Chang

Such convergence is not guaranteed, however.

CHALLENGES FOR GAN TRAINING

- Non-convergence: the model parameters oscillate, destabilize and never converge.
- Mode collapse: the generator collapses which produces limited varieties of samples.
- Diminished gradient: the discriminator gets too successful that the generator gradient vanishes and learns nothing.
- Unbalance between the generator and discriminator causing overfitting.
- Highly sensitive to the hyperparameter selections.

REFERENCES

Ian Goodfellow (2016)
NIPS 2016 Tutorial: Generative Adversarial Networks
https://arxiv.org/abs/1701.00160

Lilian Weng (2017)
From GAN to WGAN
https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Mark Chang (2016)
Generative Adversarial Networks
https://www.slideshare.net/ckmarkohchang/generative-adversarial-networks