NYU

Introduction to Data Science

Center for Data Science
Iddo Drori, Spring 2019

NYU

Bias

Word Embedding

- Replace 1-hot word representation with lower dimensional feature vector for each word.
- 1-hot representation is limited, dot product of any two 1-hot word representations is 0, so does not capture relationships among words.
- Finite dictionary of words, finite fixed encoding, embedding, of words, into lower dimensional space.
- Learn word embedding from large unsupervised text corpus.
- Use in supervised task, taking each word embedding as input.

Word Embedding Bias

- Analogies

$$A : B :: A' : B'$$

  - *Eman : Ewoman :: Eking : Equeen* (Mikolov et al 2013)

$$\underset{w}{\operatorname{argmax}}\, d(e_{B'}, e_A - e_B + e_{A'}) \qquad d(u, v) = \frac{u^T v}{\|u\|\|v\|}$$

- Fix bias in corpus
  - *Eman : Ewoman :: Eprogrammer : Ehomemaker* (Bolukbasi 2016)
  - Project onto bias direction

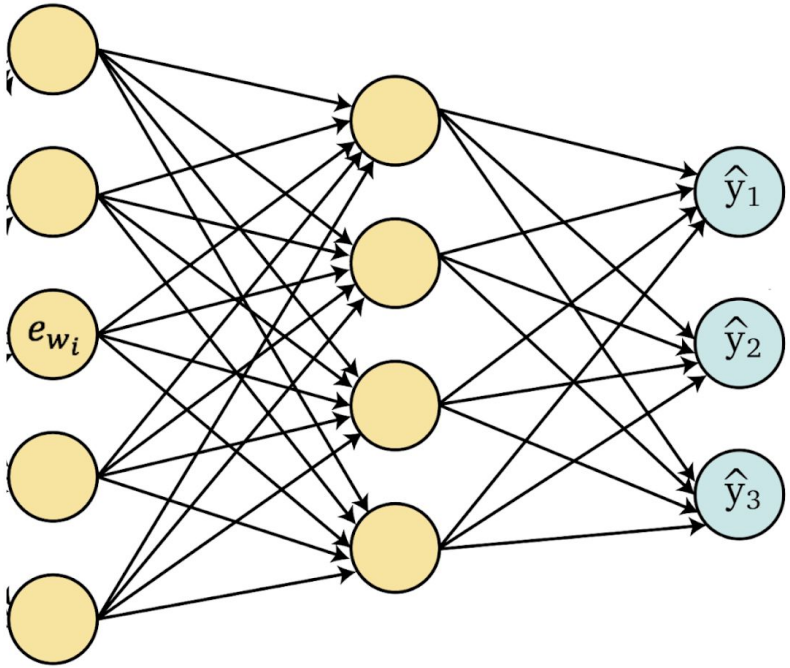
Word Embedding

$$\underset{n\times p}{E}\ \underset{p\times 1}{o_w} = \underset{n\times 1}{e_w}$$

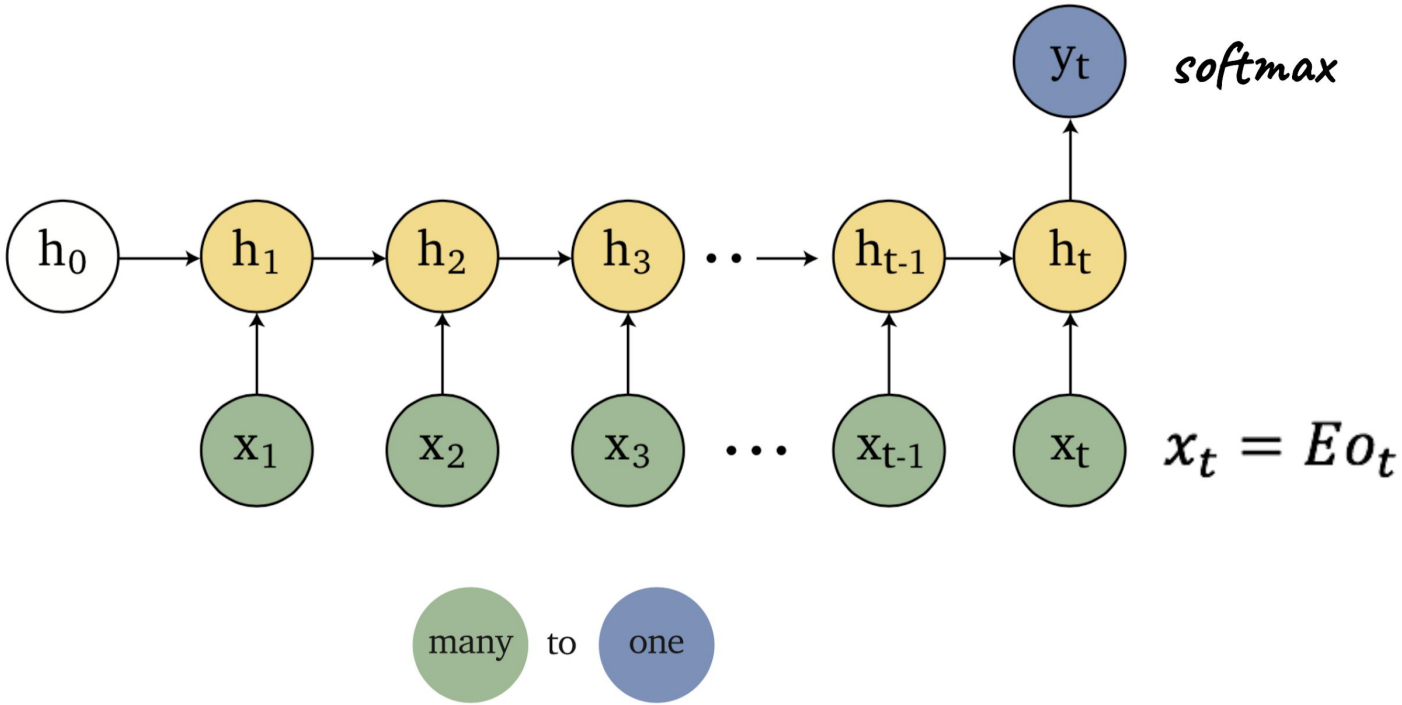
Word Embedding

Mary had a little lamb whose fleece was white as snow

Eo_{w_i}



Application



NYU

Fairness

NYU

Domains

- Bank loans
- Advertising
- University admissions

Fairness

- Individual fairness: treat similar individuals similarly
- Group fairness: statistical (demographic) parity, demographics of those receiving positive (or negative) classifications are same as demographics of entire population

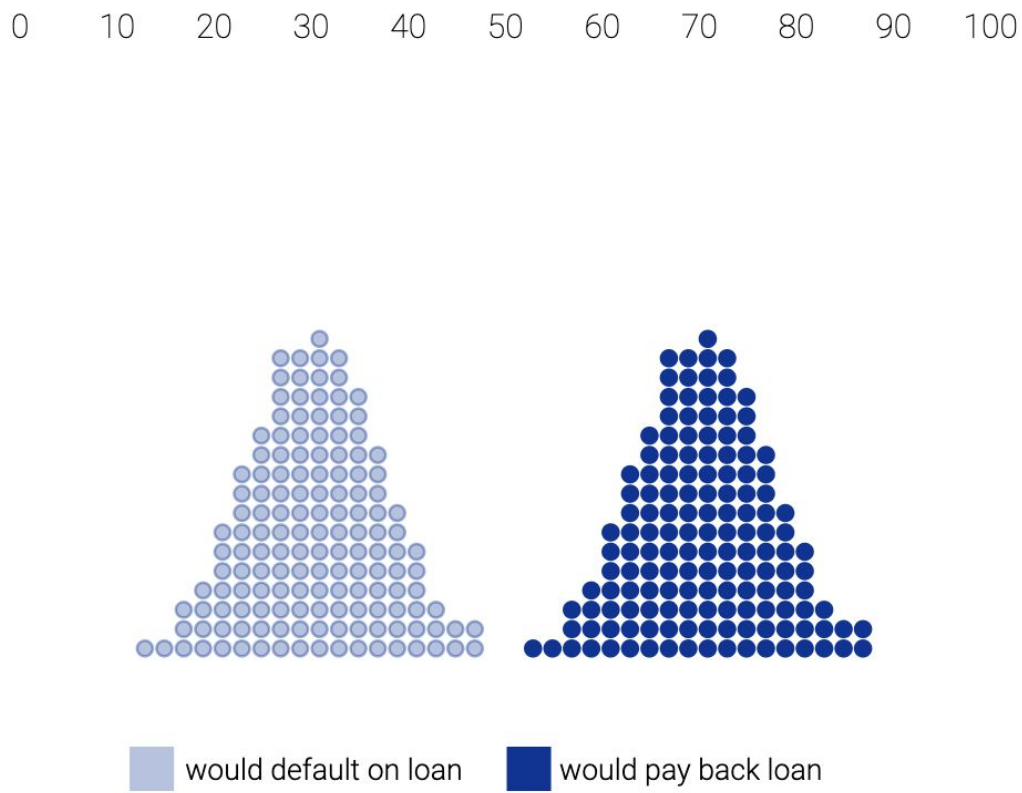
Model

- Score x
- Probability $p(x)$
- Utility $u(x)$
- Change in score $d(x)$

Loans: Model

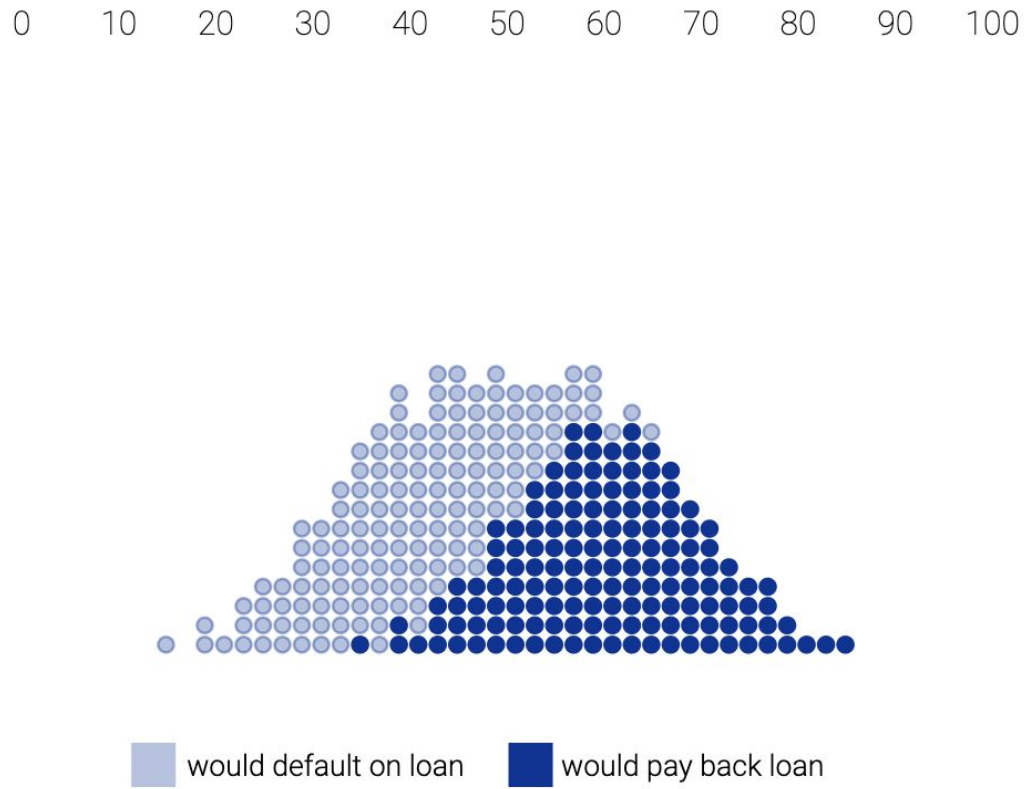
- Bank grants or denies loan based on credit score x
- Higher credit score represents higher likelihood of payback $p(x)$
- Bank selects threshold: people with score below threshold are denied, people with score above threshold are granted loan
- Bank has utility $u(x)$ which is the expected return from loan. Successful loan makes bank a profit, default costs bank
- Credit score changes $d(x)$ based on whether loan is paid back or not

Loans: Separation



Source: research.google.com/bigpicture/attacking-discrimination-in-ml

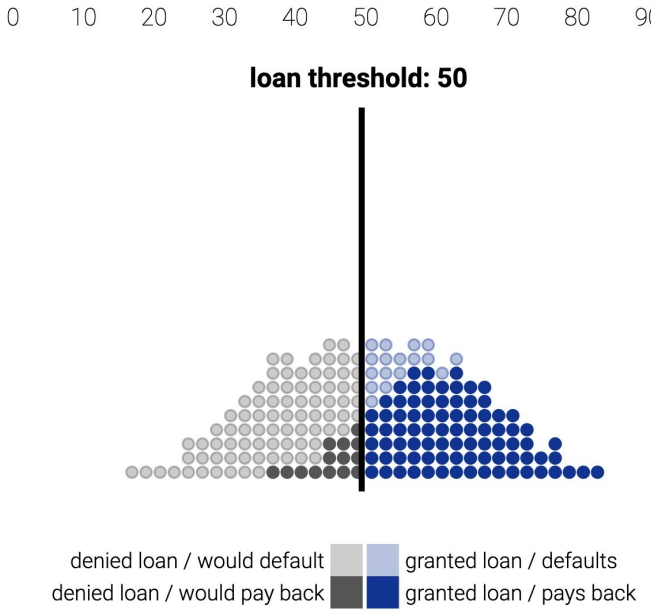
Loan Granting: Overlapping Categories



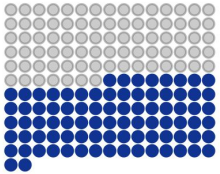
Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Decision Threshold and Outcome

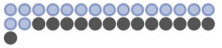
- Bank selects threshold: people with score below threshold denied, people with score above granted loan
- Successful loan makes bank $300 profit, default costs bank $700



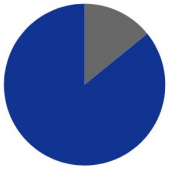
Correct 84%
loans granted to paying applicants and denied to defaulters



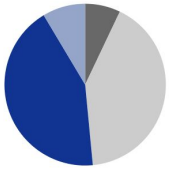
Incorrect 16%
loans denied to paying applicants and granted to defaulters



True Positive Rate 86%
percentage of paying applications getting loans



Positive Rate 52%
percentage of all applications getting loans

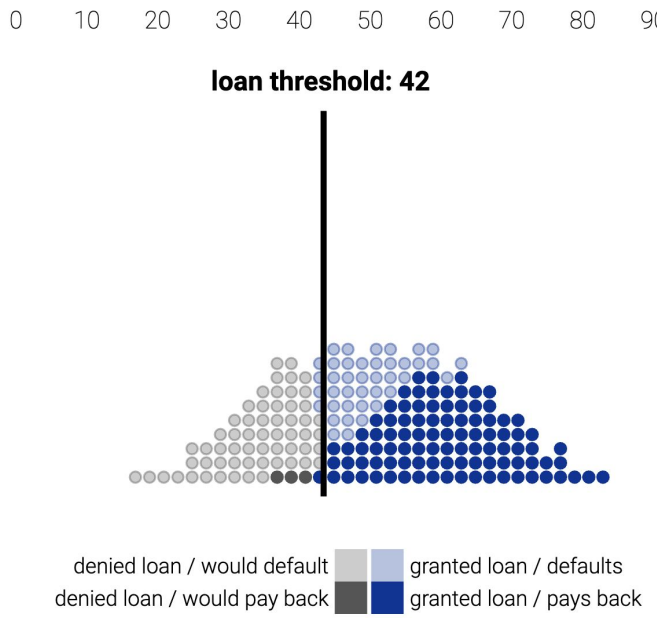


Profit: **13600**

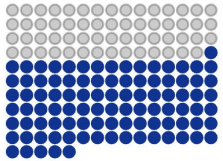
Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Decision Threshold and Outcome

- Setting threshold too low: bank gives loans to many people who default

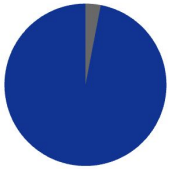


Correct 78%
loans granted to paying applicants and denied to defaulters

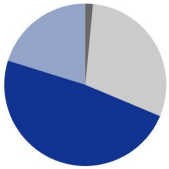


Incorrect 22%
loans denied to paying applicants and granted to defaulters

True Positive Rate 97%
percentage of paying applications getting loans



Positive Rate 69%
percentage of all applications getting loans

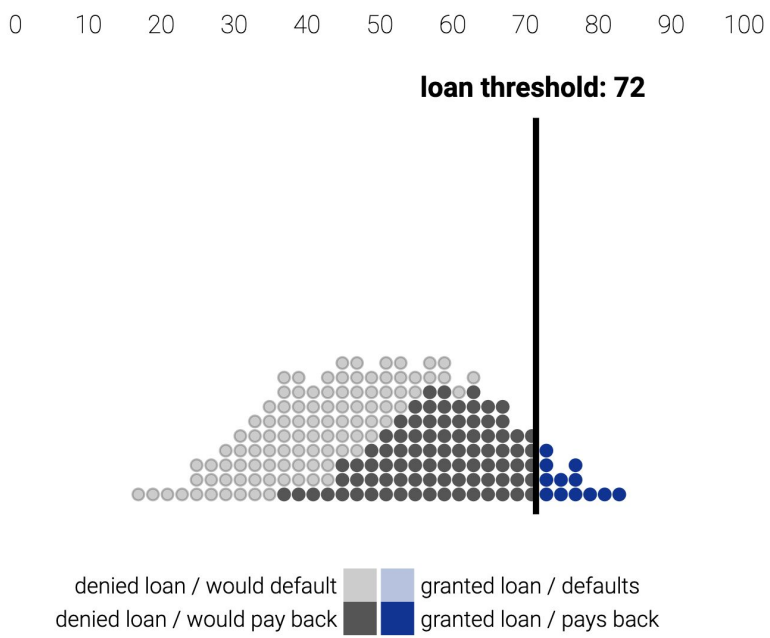


Profit: **800**

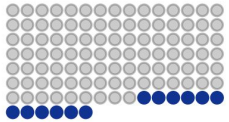
Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Decision Threshold and Outcome

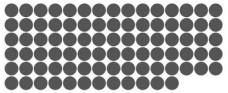
- Setting threshold too high: many people who deserve loan do not get a loan



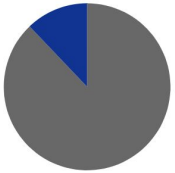
Correct 56%
loans granted to paying applicants and denied to defaulters



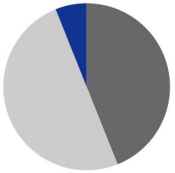
Incorrect 44%
loans denied to paying applicants and granted to defaulters



True Positive Rate 12%
percentage of paying applications getting loans



Positive Rate 6%
percentage of all applications getting loans

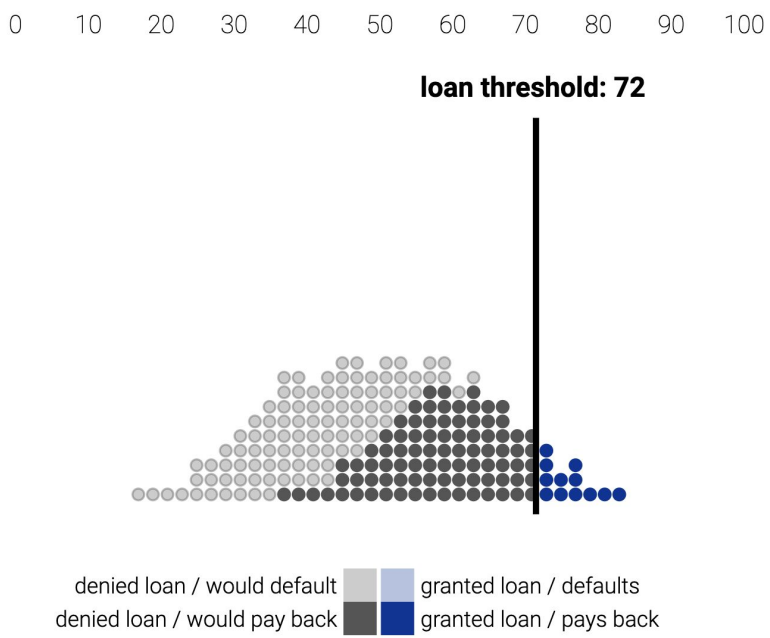


Profit: **3600**

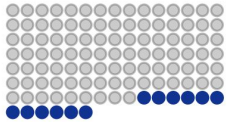
Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Decision Threshold and Outcome

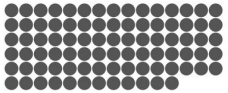
- Setting threshold too high: many people who deserve loan do not get a loan



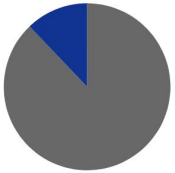
Correct 56%
loans granted to paying applicants and denied to defaulters



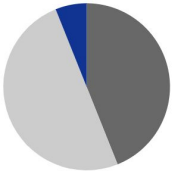
Incorrect 44%
loans denied to paying applicants and granted to defaulters



True Positive Rate 12%
percentage of paying applications getting loans



Positive Rate 6%
percentage of all applications getting loans



Profit: **3600**

Source: research.google.com/bigpicture/attacking-discrimination-in-ml

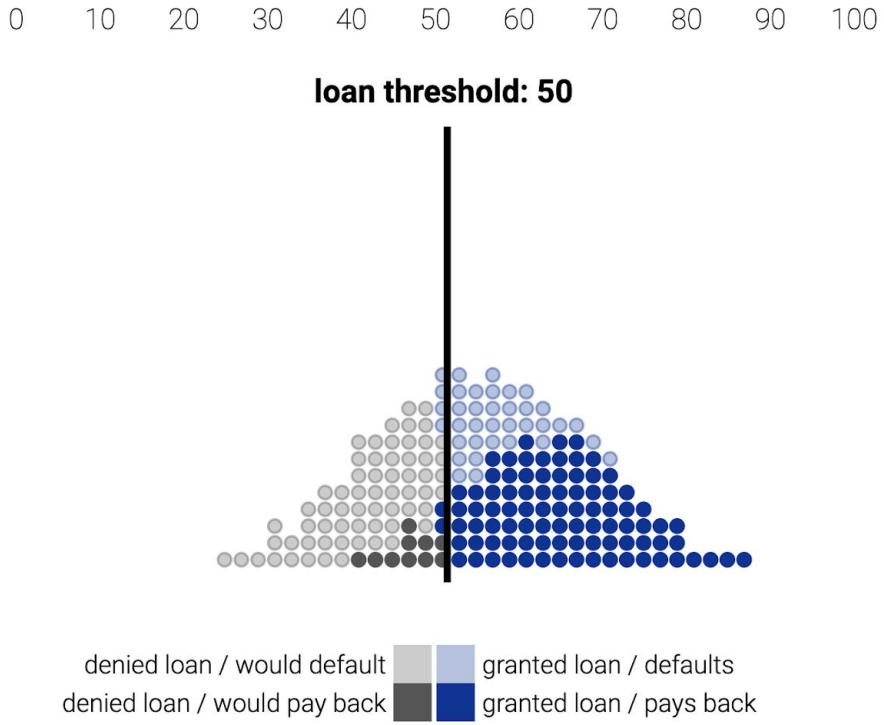
Loans: Selecting Threshold

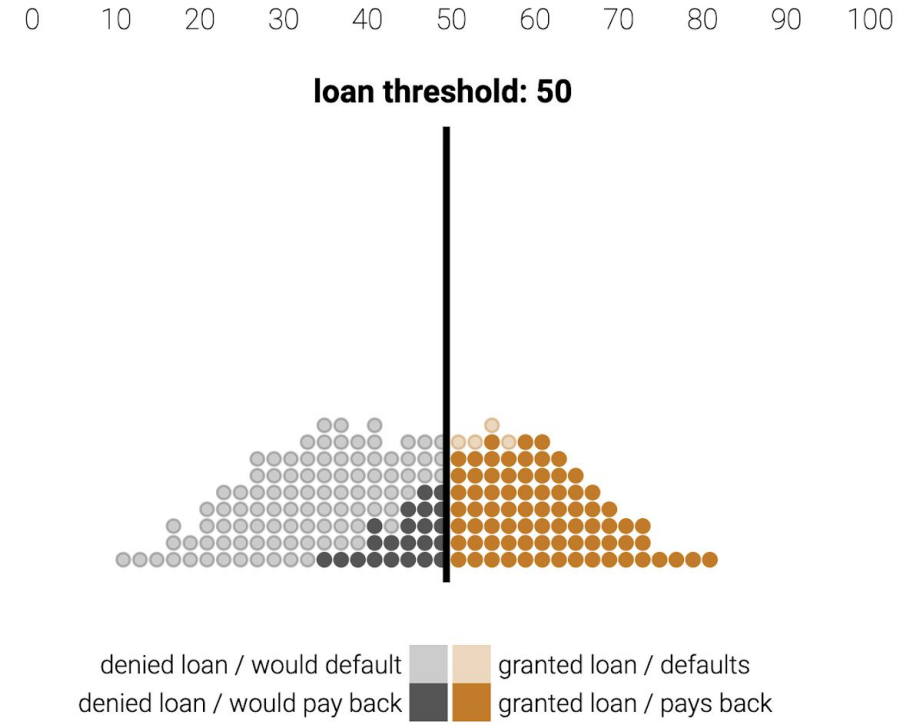
Possible criteria

- Maximize number of correct decisions
- Maximize profit
- Group unawate
- Demographic parity
- Equal opportunity

Loans: Same Threshold

- Two groups of people in population: blue and orange, equally likely to payback loan

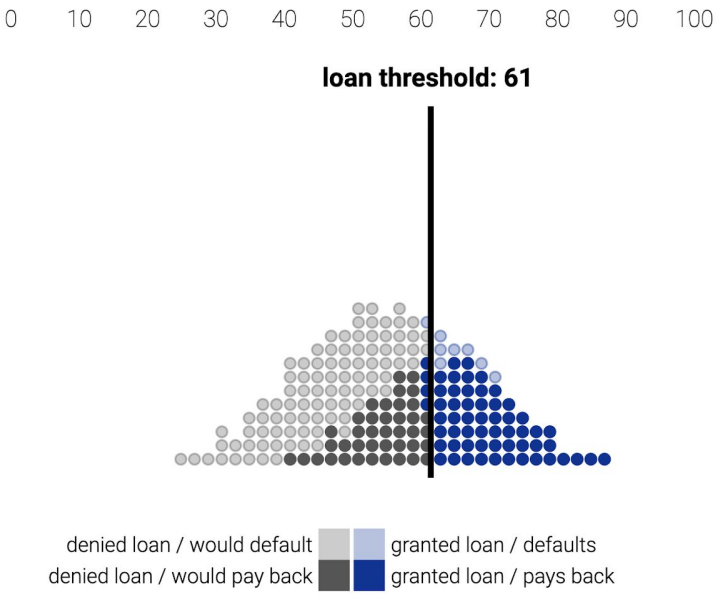


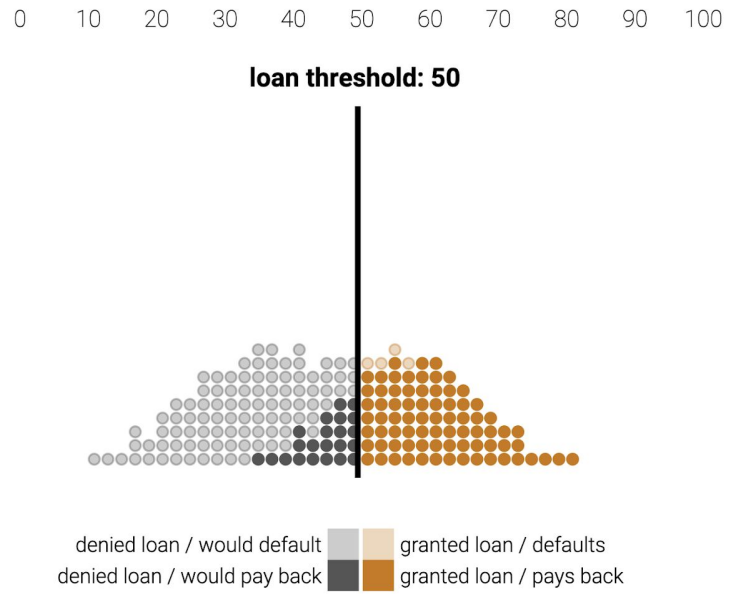


Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Maximize Profit

- Problem: higher threshold for blue group than orange, even though groups equally likely to pay back loan



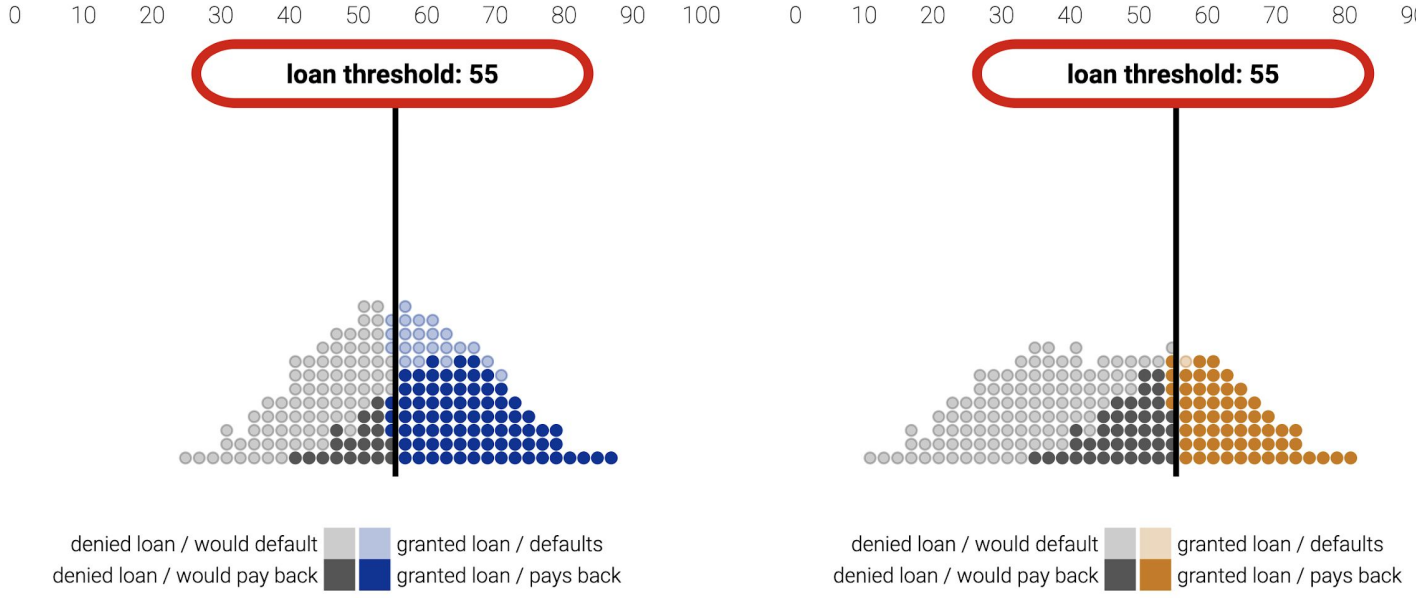


Total profit = **32400**

Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Group Unaware

- Maximize profit while holding groups to same threshold
- Problem: both groups equally loan-worthy, difference in score distributions, orange group gets fewer loans

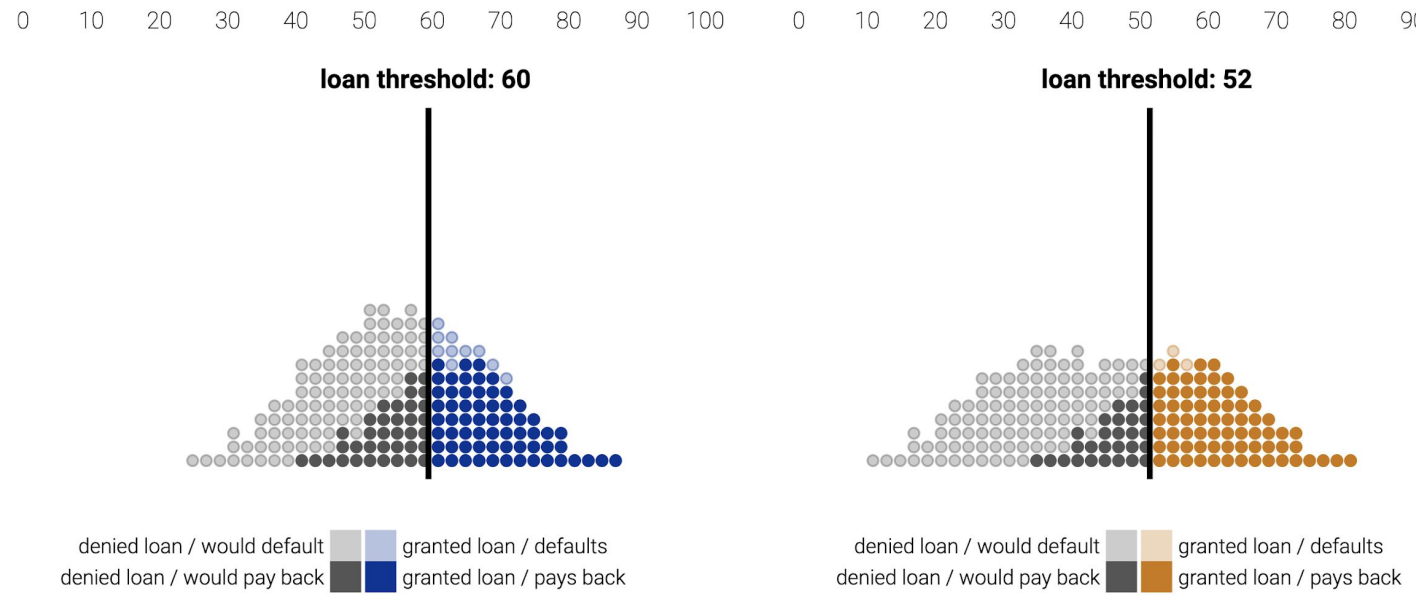


Total profit = **25600**

Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Demographic Parity

- Groups receive same fraction of loans

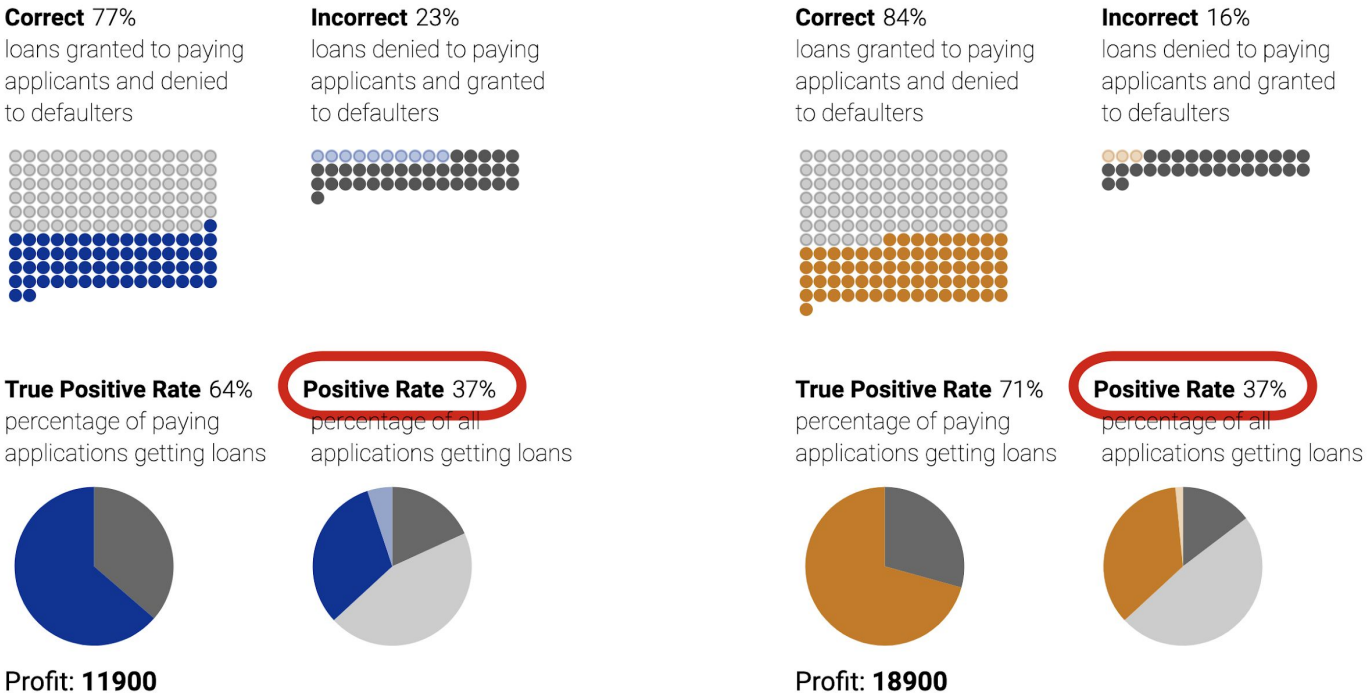


Total profit = **30800**

Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Demographic Parity

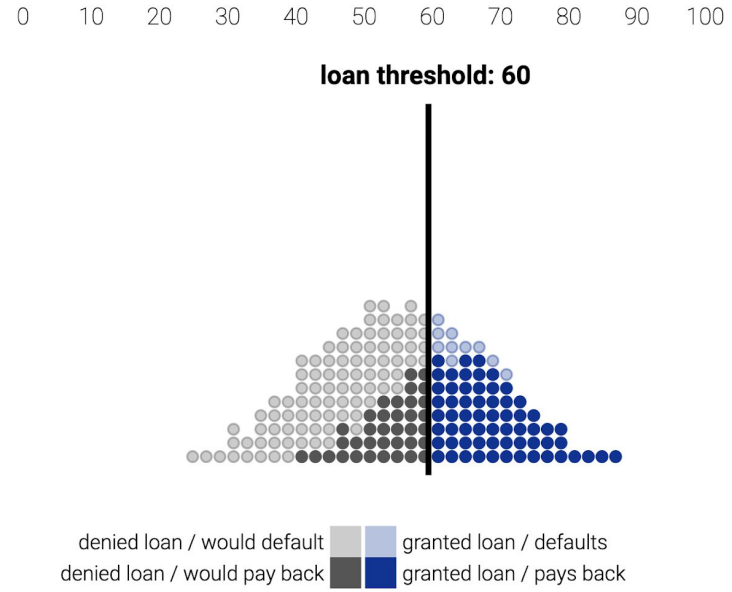
- Loan thresholds set so same fraction of loans to each group, dsmr positive rate for both groups

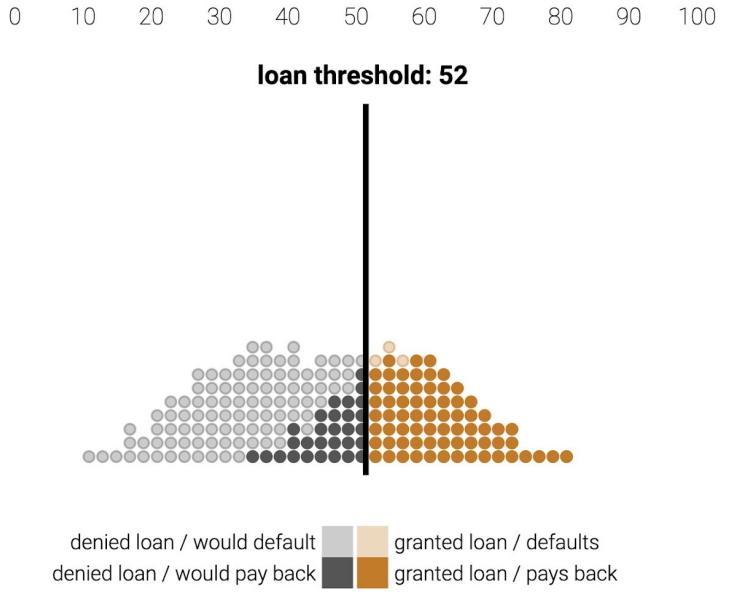


Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Demographic Parity

- Problem: only looks at loans granted, not rates at which loans paid back.
- Fewer qualified people in blue group granted loans than in orange group



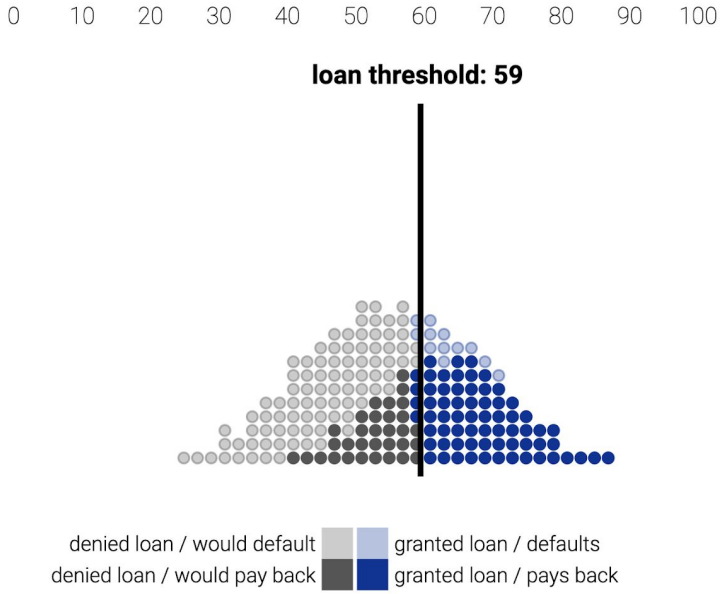


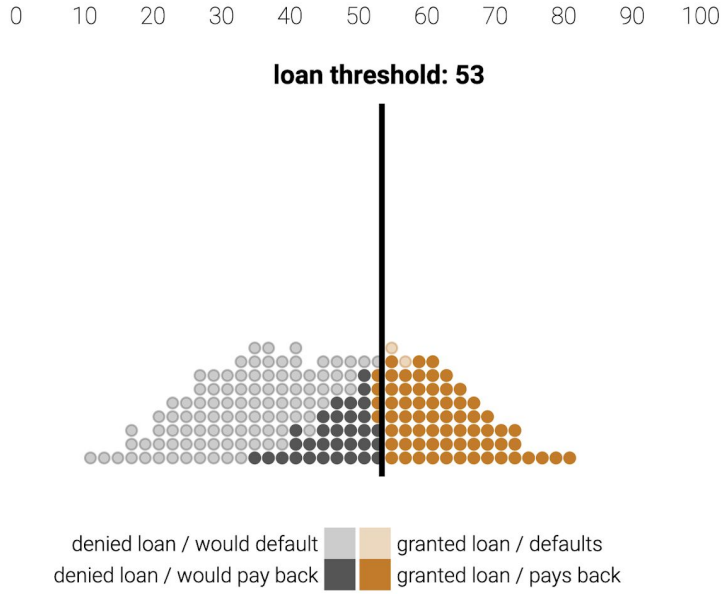
Total profit = **30800**

Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Equal Opportunity

- Of people who pay back loan, same fraction in each group actually granted loan.
- Same true positive rate for both groups



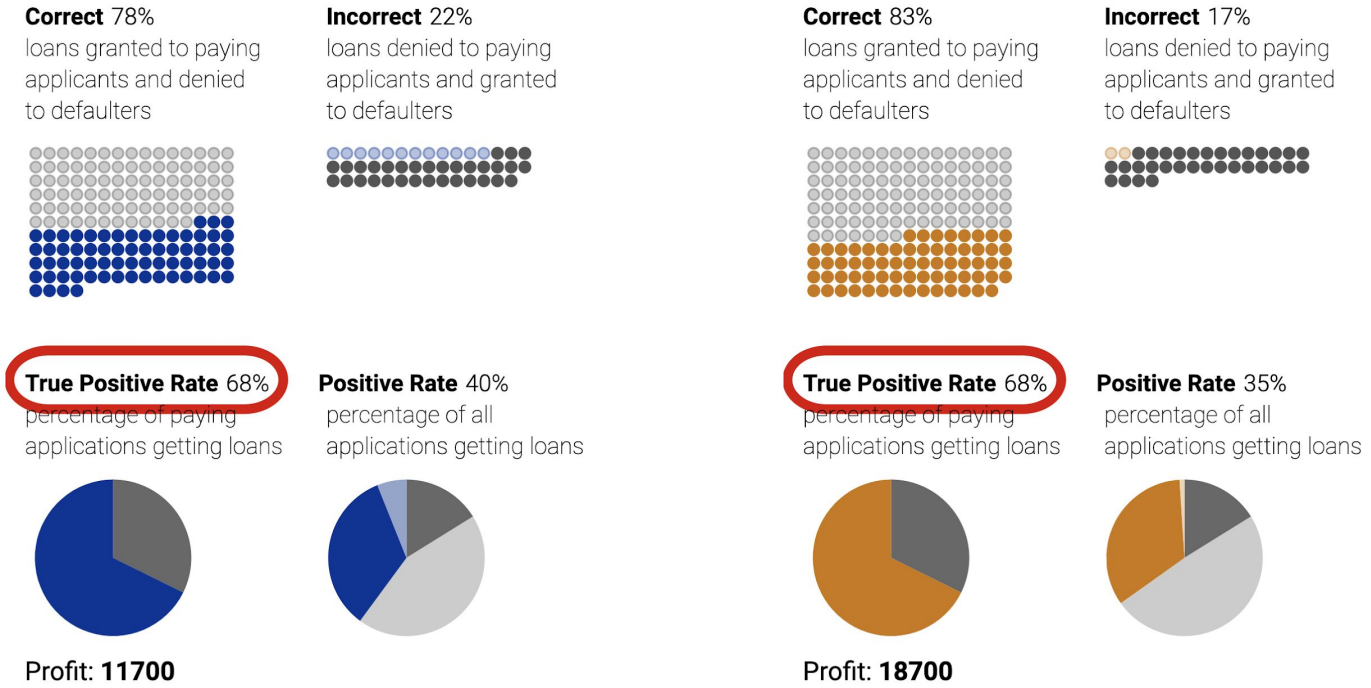


Total profit = **30400**

Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Loans: Equal Opportunity

- Of people who pay back loan, same fraction in each group actually granted loan.
- Same true positive rate for both groups



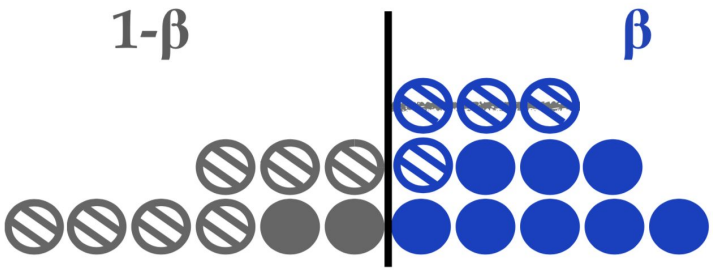
Source: research.google.com/bigpicture/attacking-discrimination-in-ml

Equal Error Rates Across Groups

- False positive rate $FPR = FP / N = FP / (FP + TN)$
- False negative rate $FNR = FN / P = FN / (FN + TP)$
- Positive predictive value $PPV = TP / (TP + FP)$
- No classifier can ensure all 3 criteria together unless equal base rates

Loans: Acceptance Rate

- In [0,1]

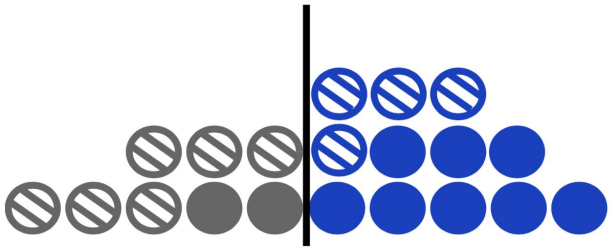


Source: Delayed impact of fair machine learning, Liu et al, ICML 2018

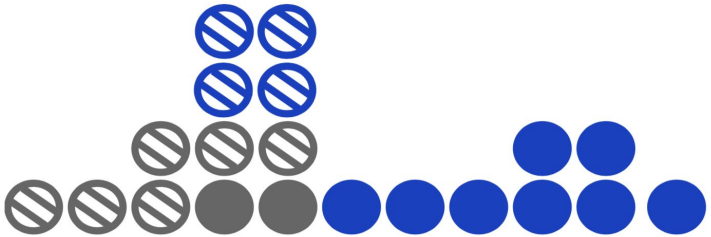
Loans: Delayed Impact

- Scores change with pay back or default
  if pay back: new score = old score + c
  If default: new score = old score - c
  delayed impact = mean(new score - old score) of group

- Before



- After

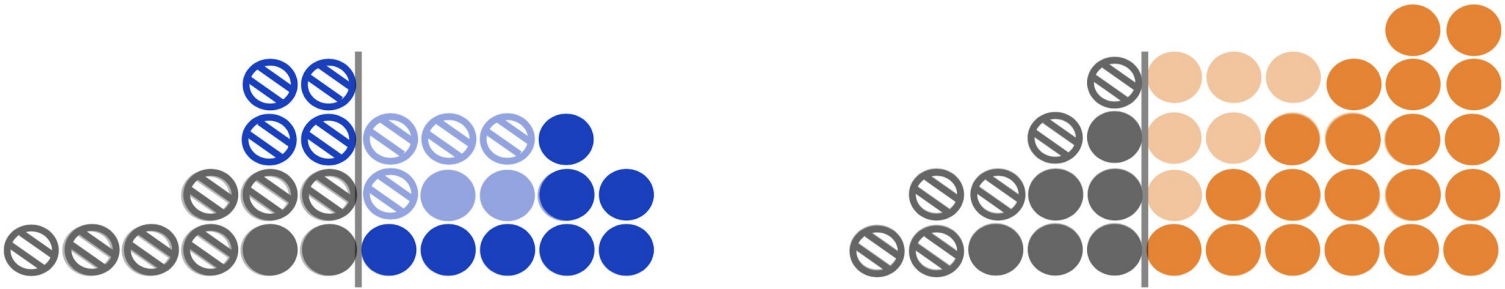


Source: Delayed impact of fair machine learning, Liu et al, ICML 2018

Loans: Demographic Parity Delayed Impact

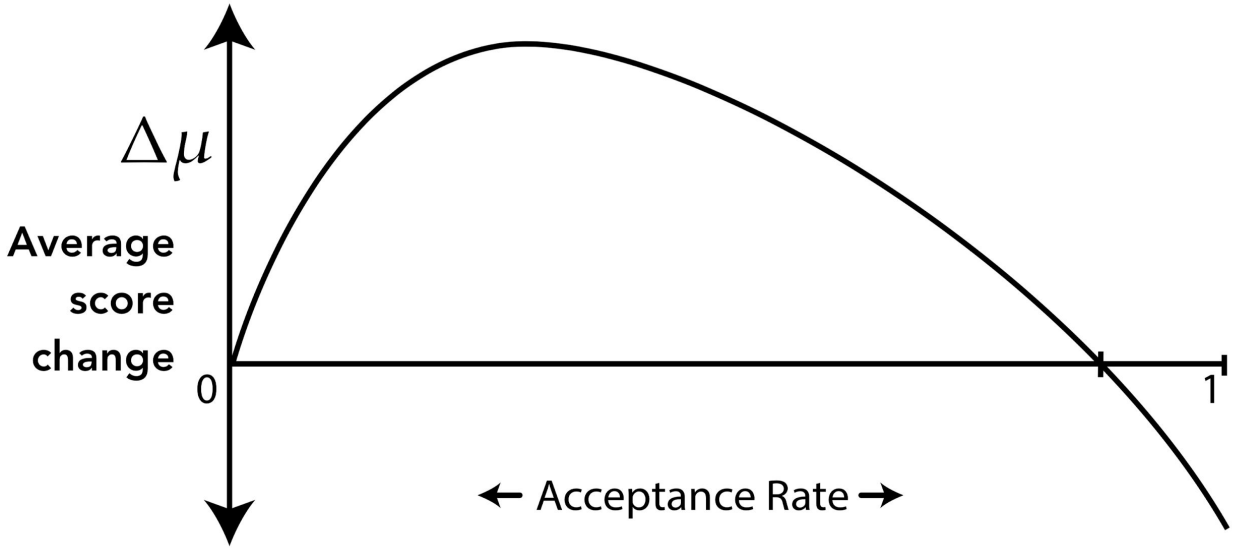
- Credit scores change with pay back or default, harming blue group and benefitting orange group
- Before

- After



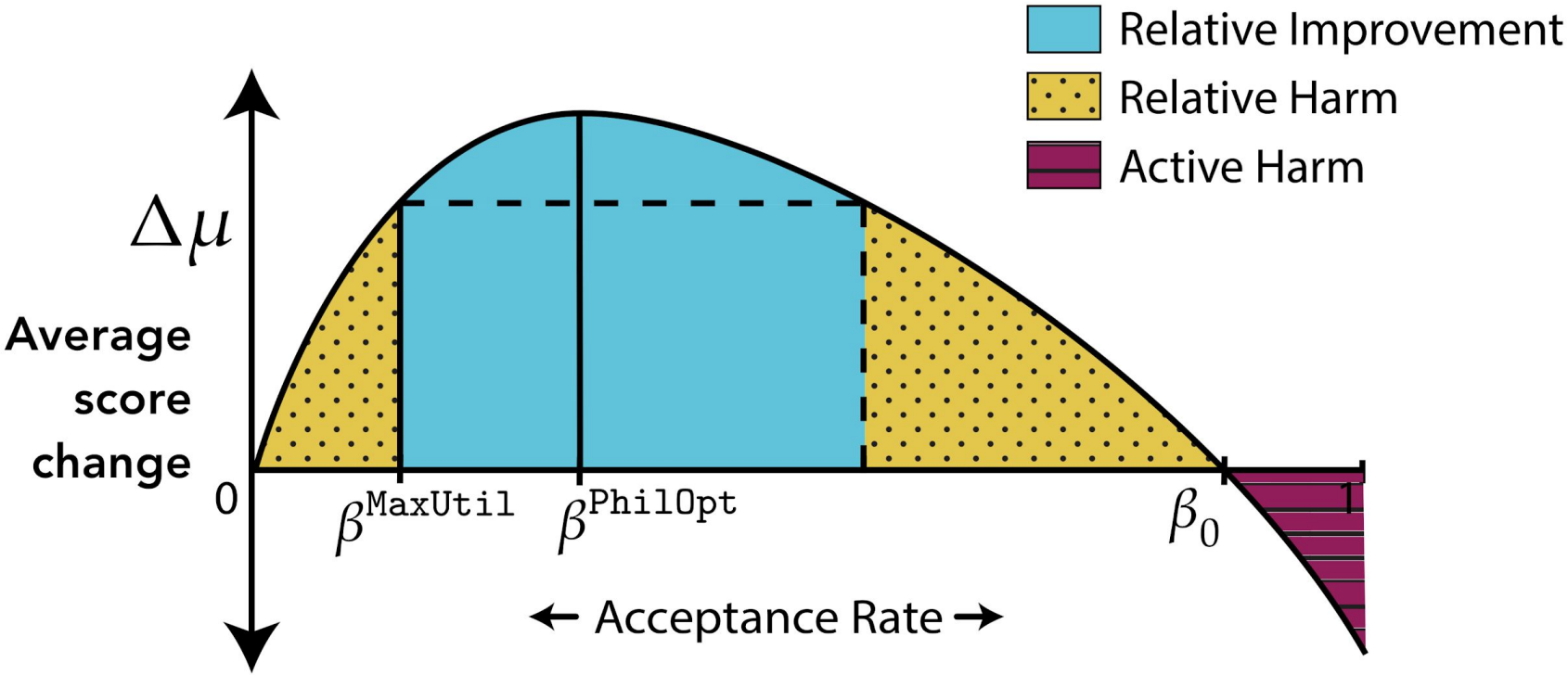
Source: Delayed impact of fair machine learning, Liu et al, ICML 2018

Loans: Delayed Impact Outcome Curve



Source: Delayed impact of fair machine learning, Liu et al, ICML 2018

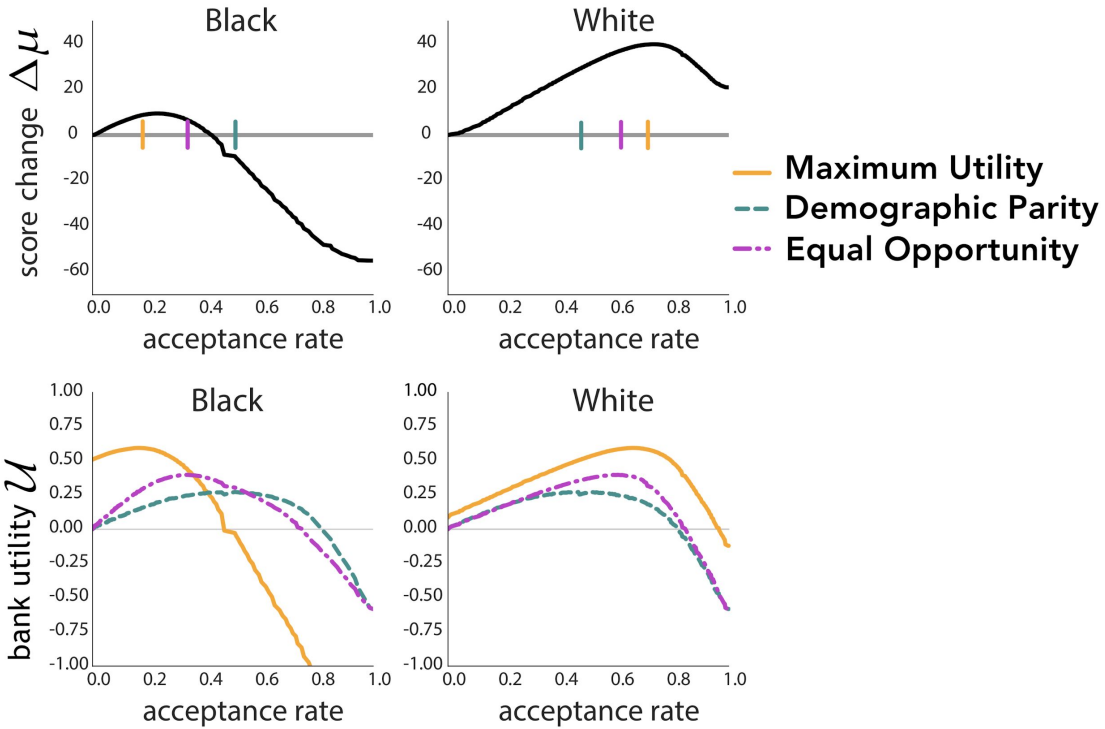
Loans: Delayed Impact Outcome Curve



Source: Delayed impact of fair machine learning, Liu et al, ICML 2018

Loans: Delayed Impact Outcome Curves

- 300,000+ TransUnion TransRisk scores from 2003



Source: Delayed impact of fair machine learning, Liu et al, ICML 2018

Advertising: Model

- Agencies decide who to target
- People have product interest scores x
- Scores represent probability of positive response to ad $p(x)$
- Agency have utility $u(x)$ of targeting, which increases with score
- People seeing ad that are not interested may react negatively reducing interest in product, reducing their score by $d(x)$

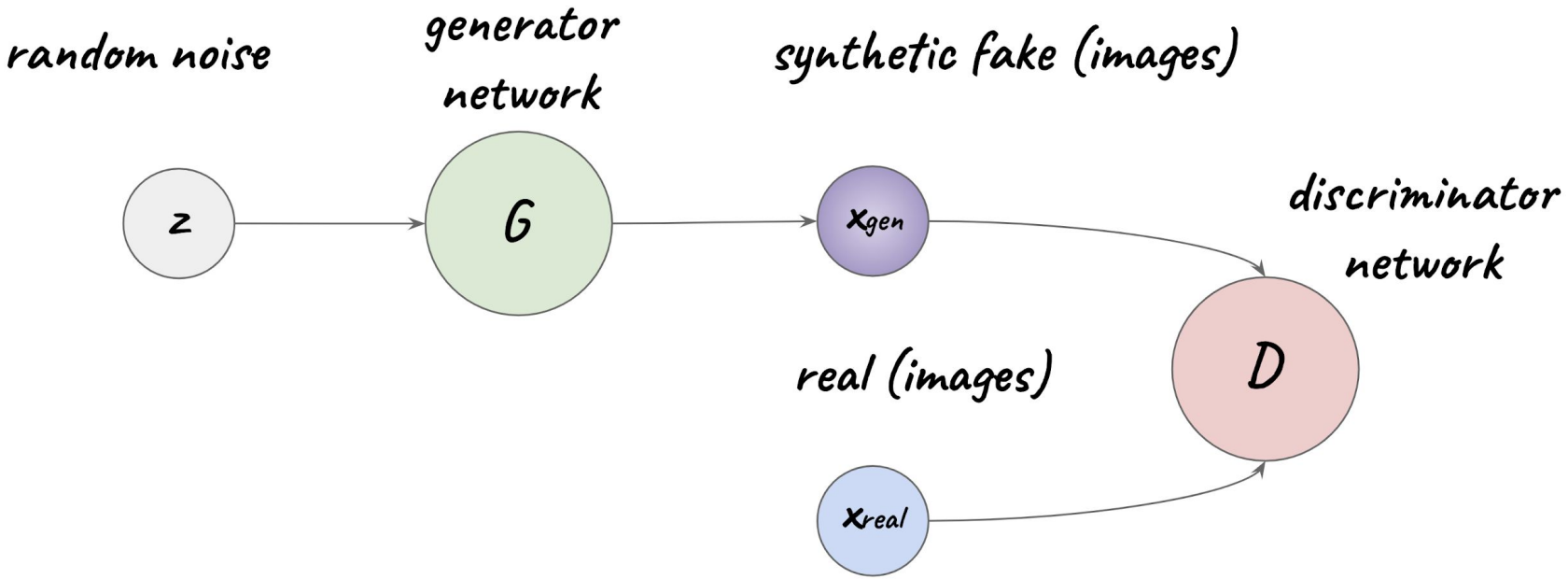
University Admissions: Model

- Students have university preparedness score x
- Students admitted succeed with probability based on their score $p(x)$
- University has utility $u(x)$ from alumni donations, positive ratings when student succeeds, and loss if student fails.
- Students success in university effects later success, changing score by $d(x)$

NYU

Generative Adversarial Networks

Generative Adversarial Networks

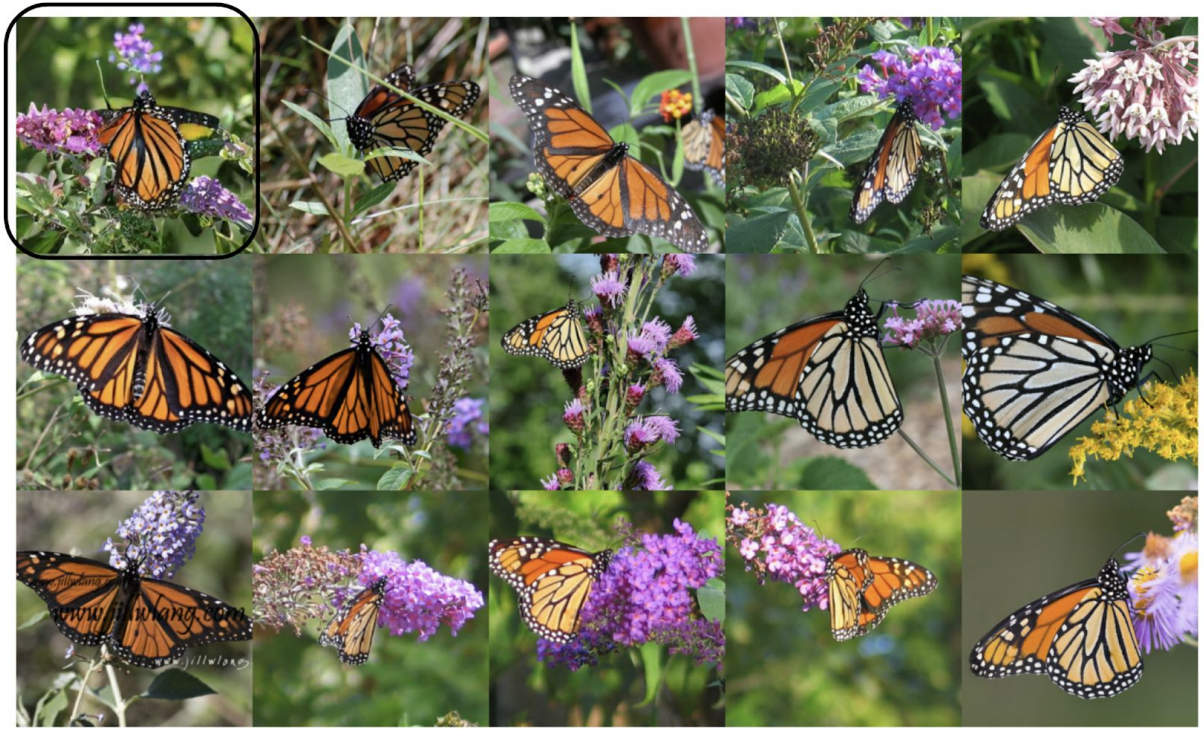


G tries to synthesis fake examples that fool D

D tries to identify the fakes

GANs

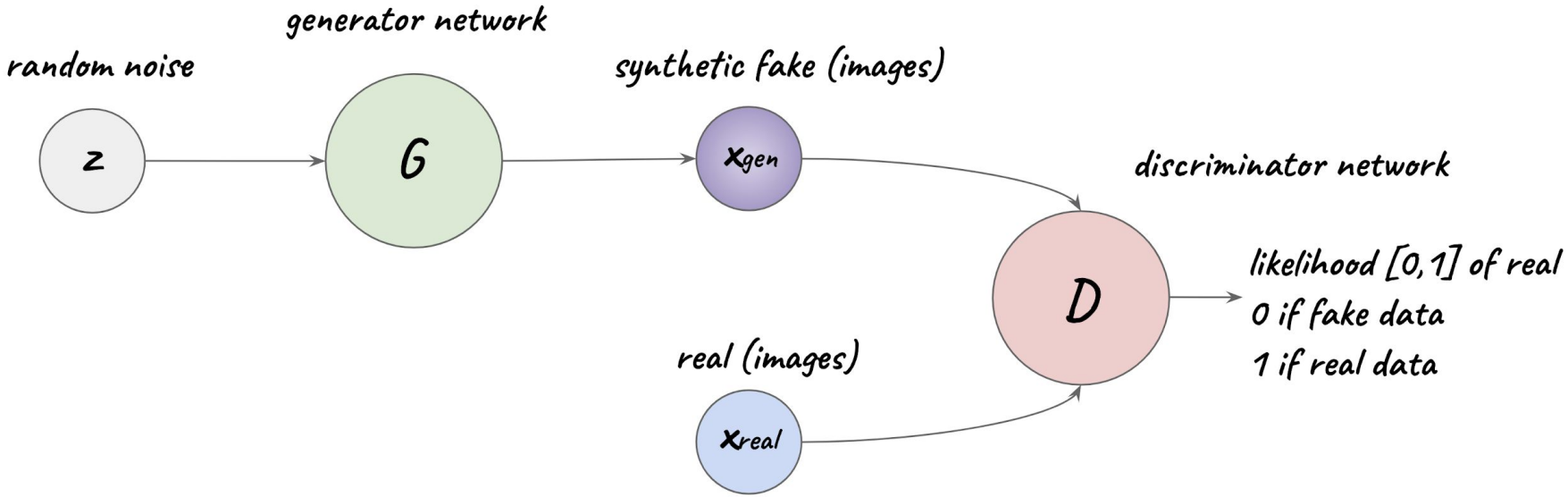
synthesized



all the rest are nearest neighbors in training set

Source: Large scale GAN training for high fidelity natural image synthesis, Brock et al 2019.

GAN Objectives

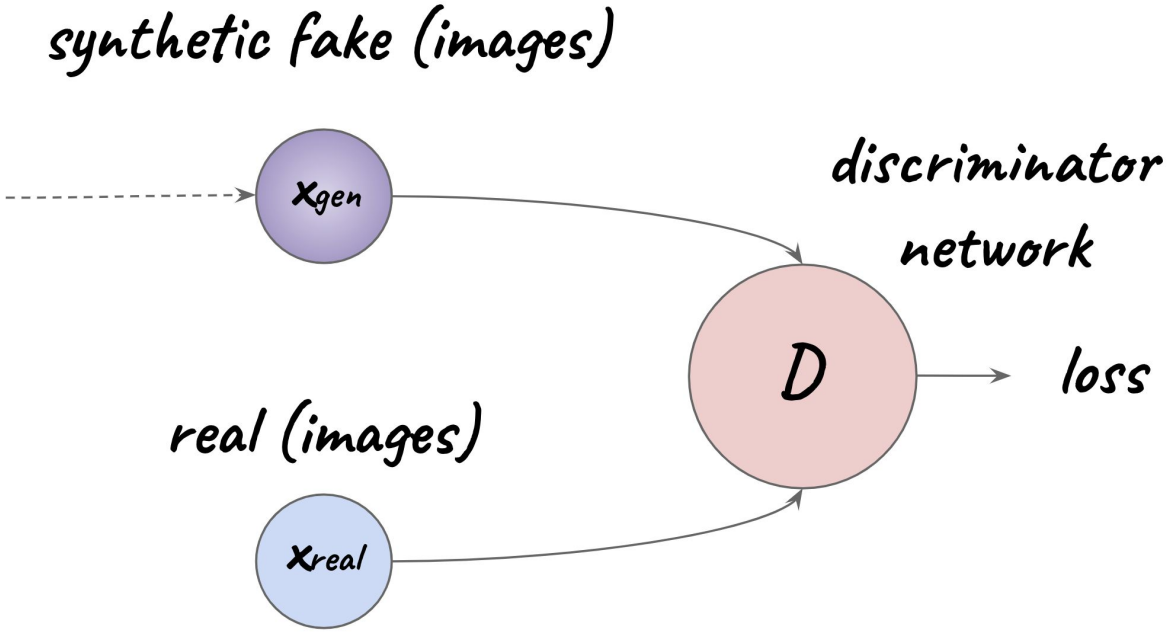


If x is real D(x) -> 1

If x=G(z) is fake then D tries to make D(G(z)) -> 0

whereas G tries to make D(G(z)) -> 1

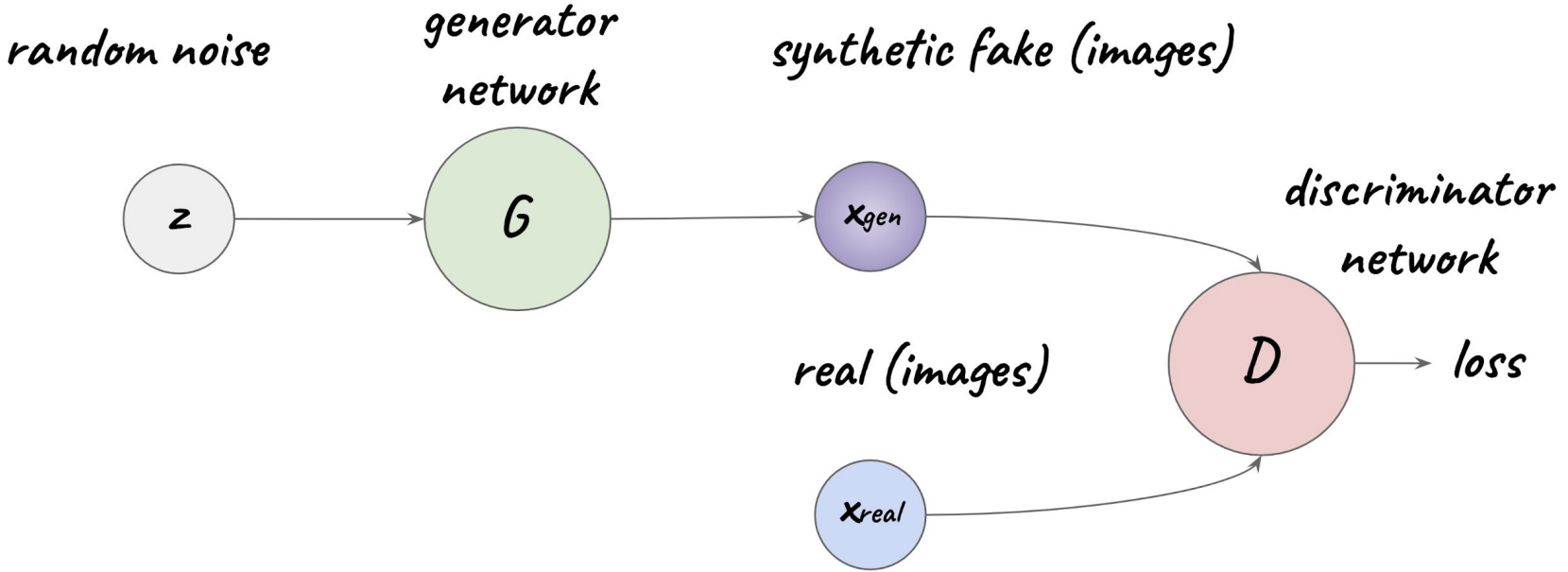
GAN Discriminator



D tries to distinguish between real and fake

optimizing weights θ_D

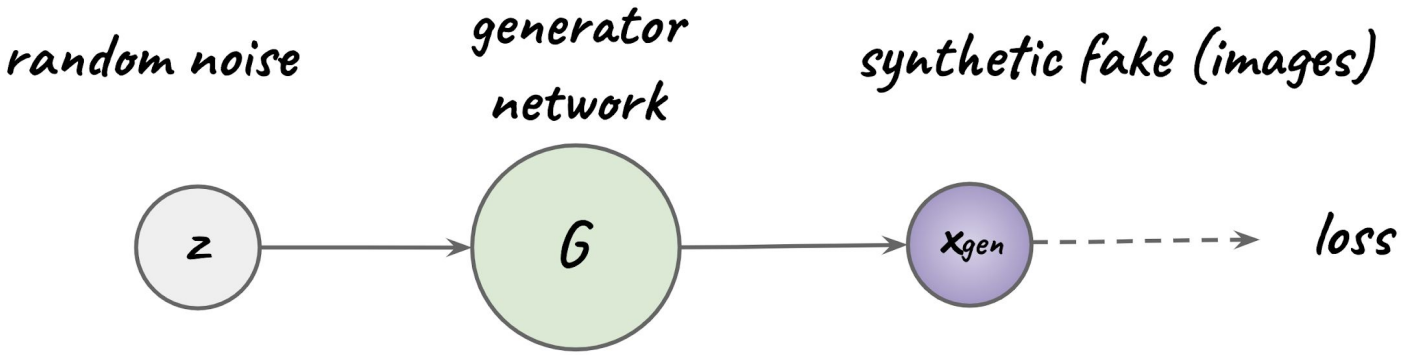
GAN Generator



G tries to synthesize fake examples that fool D

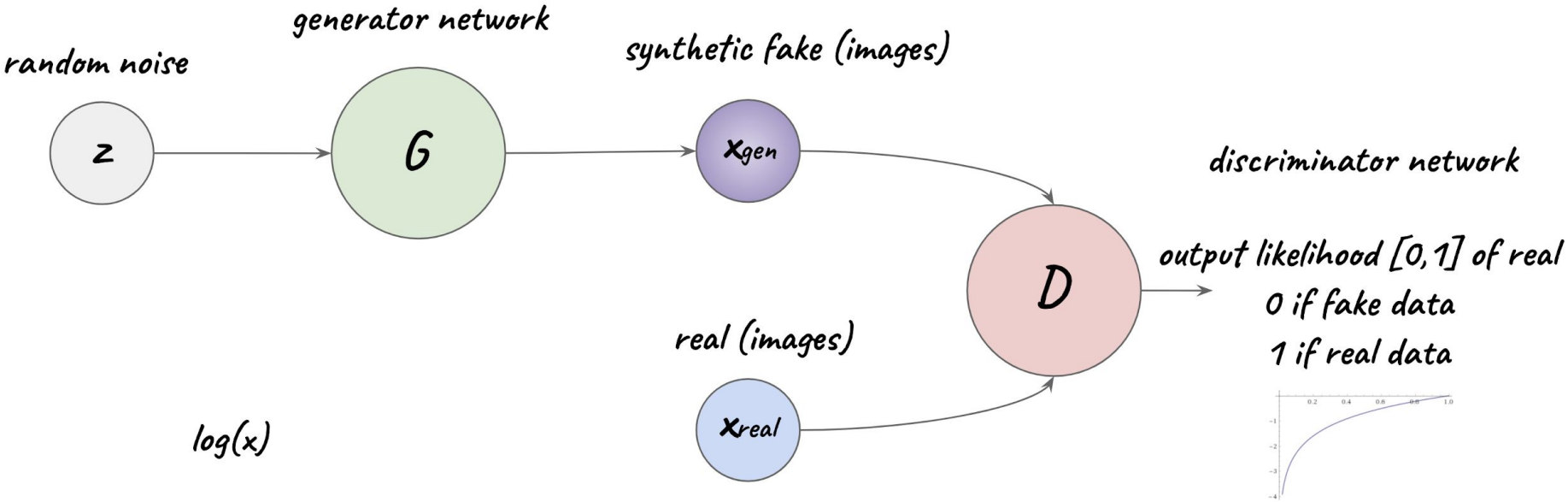
optimizing weights θ_G

GAN Generator



from generators perspective D is a loss function to be maximized
rather than being hand design it is learnt

GAN Objective



$$\underset{\theta_G}{\text{minimize}}\ \underset{\theta_D}{\text{maximize}} \left(\mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D}\left(G_{\theta_G}(z) \right) \right) \right)$$

discriminator output for real data x

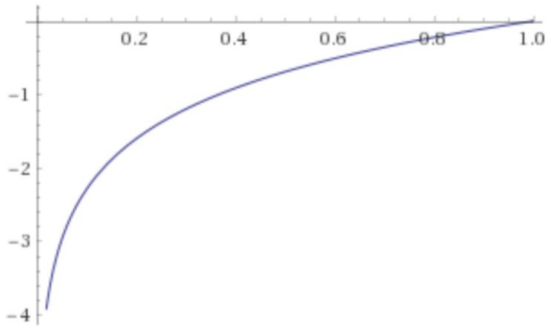
discriminator output for fake data G(z)

GAN Objective

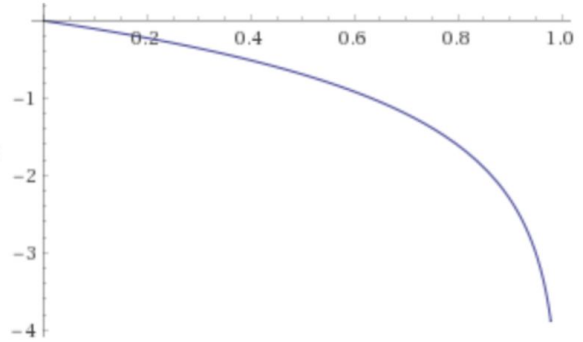
discriminator network outputs likelihood [0,1] of being real

0 if fake data
1 if real data

log(x)



log(1-x)



if x is real then discriminator goal is D(x)->1

if x=G(z) is fake then D tries to make D(G(z)) -> 0
whereas G tries to make D(G(z)) -> 1

$$\underset{\theta_G}{\text{minimize}}\ \underset{\theta_D}{\text{maximize}} \left(\mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D}\left(G_{\theta_G}(z) \right) \right) \right)$$

discriminator output for real data x

discriminator output for fake data G(z)

GAN Objective

- Training as a 2-player game: saddle point problem

$$\underset{\theta_G}{\text{minimize}}\ \underset{\theta_D}{\text{maximize}} \left(\mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

Iteratively alternate between

1. Gradient ascent on discriminator: cross-entropy objective

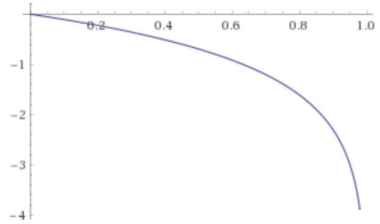
$$\underset{\theta_D}{\text{maximize}} \left(\mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

2. Gradient descent on generator

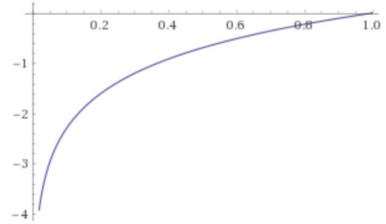
$$\underset{\theta_G}{\text{minimize}} \left(\mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

GAN Generator Objective

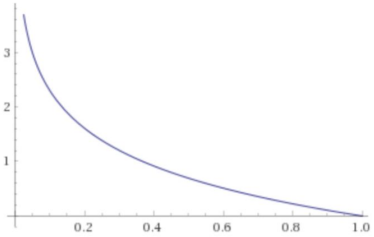
$$\underset{\theta_G}{\text{minimize}} \left(\mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$



$$\underset{\theta_G}{\text{maximize}} \left(\mathbb{E}_{z \sim p(z)} \log \left(D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

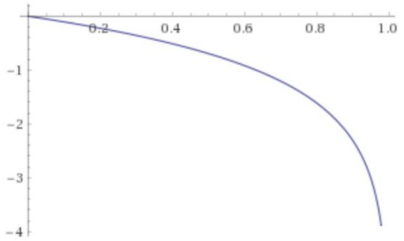


$$\underset{\theta_G}{\text{minimize}} \left(\mathbb{E}_{z \sim p(z)} -\log \left(D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$



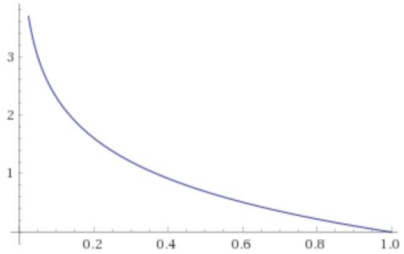
GAN Generator Objective

$$\underset{\theta_G}{\text{maximize}} \frac{1}{m_G} \sum_{i=1}^{m_G} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$



saturating cost

$$\underset{\theta_G}{\text{minimize}} -\frac{1}{m_G} \sum_{i=1}^{m_G} \log D\left(G\left(z^{(i)}\right)\right)$$



non-saturating cost

GAN Objective

- Training as a 2-player game: saddle point problem

$$\underset{\theta_G}{\text{minimize}}\ \underset{\theta_D}{\text{maximize}} \left(\mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

Iteratively alternate between

1. Gradient ascent on discriminator: cross-entropy objective

$$\underset{\theta_D}{\text{maximize}} \left(\mathbb{E}_{x \sim p_{data}} \log D_{\theta_D}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

2. Gradient ascent on generator on different objective

$$\underset{\theta_G}{\text{maximize}} \left(\mathbb{E}_{z \sim p(z)} \log \left(D_{\theta_D} \left(G_{\theta_G}(z) \right) \right) \right)$$

GAN Algorithm

For n iterations

    For k iterations

        Update Discriminator

    Update Generator

GAN Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Generative adversarial nets (GANs), Goodfellow et al., NIPS 2014.

NYU

Fair Representations

Fair Representations

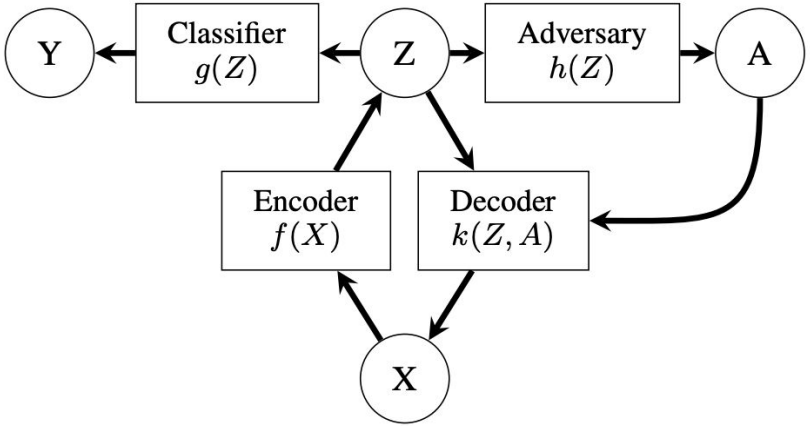
- Decision must not favour particular group, make decisions without discrimination
- Representation of data must not have identifying information, remove private information
- Adversary tries to predict relevant sensitive variables from representation

Fair Representations

- Data X
- Sensitive bit A
- Representation Z
- Labels Y

- Classifier: maps representation Z to label Y
- Adversary: maps representation Z to sensitive bit A

Fair Representations



$$L_{Adv}(h(f(X, A)), A)$$

$$\underset{f,g,k}{\text{minimize}} \; \underset{h}{\text{maximize}} \; \mathbb{E}_{X,Y,A}\left[L(f, g, h, k)\right]$$

$$\begin{aligned} L(f, g, h, k) = \alpha L_C(g(f(X, A)), Y) \\ + \beta L_{Dec}(k(f(X, A), A), X) \\ + \gamma L_{Adv}(h(f(X, A)), A) \end{aligned}$$

Source: Learning adversarially fair and transferable representations, Madras et al, 2018

Fair Representations

- Demographic parity
  Average absolute difference on each sensitive group

$$L_{Adv}^{DP}(h) = 1 - \sum_{i \in \{0,1\}} \frac{1}{|\mathcal{D}_i|} \sum_{(x,a) \in \mathcal{D}_i} |h(f(x,a)) - a|$$

- Equal opportunity
  Average absolute difference on each sensitive group-label combination

$$\mathcal{D}_i^j = \{(x,y,a) \in \mathcal{D} | a = i, y = j\}$$

$$L_{Adv}^{EO}(h) = 2 - \sum_{(i,j) \in \{0,1\}^2} \frac{1}{|\mathcal{D}_i^j|} \sum_{(x,a) \in \mathcal{D}_i^j} |h(f(x,a)) - a|$$

Source: Learning adversarially fair and transferable representations, Madras et al, 2018

Task Competitive Fairness

- Tasks compete for individuals: advertisers compete for ad slots
  Goal: individual fairness for multiple types of ads simultaneously
  Problem: higher paying advertiser leaves other type ad to rest of population

- Process
  Fix a probability distribution X over tasks
  Choose task T ~ X from probability distribution
  Classify using fair classifier for T

- Individual fairness
  For every task T and every u,v
  P(system classifies u positively for T) = P(T chosen) P(C(u) = 1)