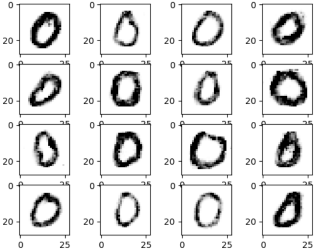
Implicit Generative Models

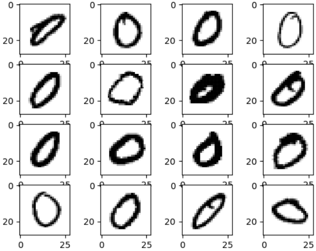
Cem Subakan

University of Illinois at Urbana-Champaign
CS598PS Guest Lecture 2

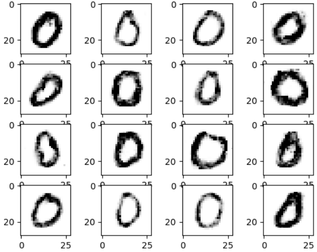
November 17'th, 2017

real or fake?

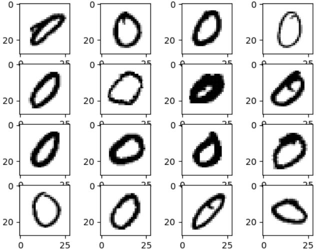




real or fake?

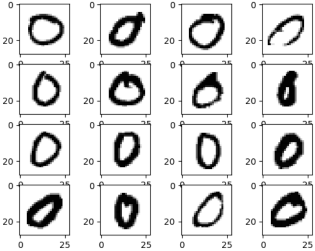


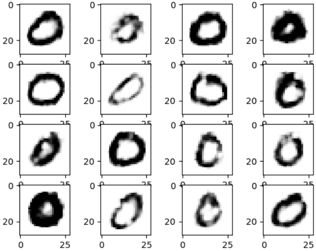
fake



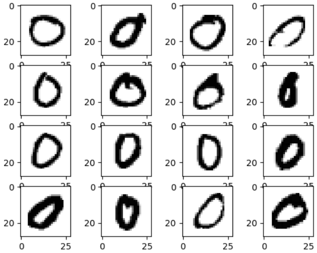
real

real or fake?

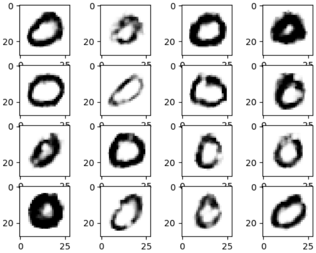




real or fake?



real



fake

Outline

Implicit Generative Models

Training Implicit Generative Models

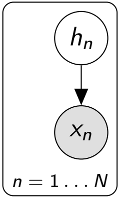
- Moment Matching
- Ratio Estimation

Plan

Implicit Generative Models

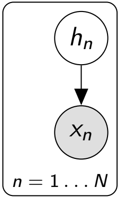
Training Implicit Generative Models
- Moment Matching
- Ratio Estimation

Generative Models



$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim p(x|h,\theta) = p_{out}(x; f_\theta(h))
\end{aligned}
$$

Generative Models

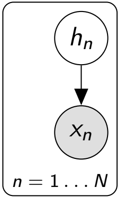


$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim p(x|h,\theta) = p_{out}(x; f_\theta(h))
\end{aligned}
$$

- Maximum Likelihood learning:

$$
\begin{aligned}
&\max_\theta \mathbb{E}_x[\log p(x|\theta)] \\
\approx &\max_\theta \sum_n \log p(x_n|\theta) \\
= &\max_\theta \sum_n \log \sum_{h_n} p(x_n, h_n|\theta) = \max_\theta \sum_n \log \sum_{h_n} p(x_n; f_\theta(h_n))
\end{aligned}
$$

Generative Models



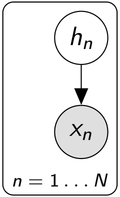
$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim p(x|h,\theta) = p_{out}(x; f_\theta(h))
\end{aligned}
$$

- Maximum Likelihood learning:

$$
\begin{aligned}
& \max_\theta \mathbb{E}_x[\log p(x|\theta)] \\
\approx & \max_\theta \sum_n \log p(x_n|\theta) \\
= & \max_\theta \sum_n \log \sum_{h_n} p(x_n, h_n|\theta) = \max_\theta \sum_n \log \sum_{h_n} p(x_n; f_\theta(h_n))
\end{aligned}
$$

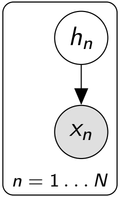
- Major problem with this:
  - What is $p(x|h,\theta)$? (Gaussian, Poisson, Smaragdisian?, Me-ian?)

Implicit Generative Model



$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim \delta(x - f_\theta(h))
\end{aligned}
$$

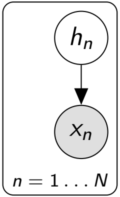
Implicit Generative Model



$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim \delta(x - f_\theta(h))
\end{aligned}
$$

- But what to do with this? What is this?
- $\delta(x - t) = \begin{cases} \infty & x = t \\ 0 & else \end{cases}.$

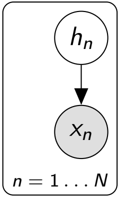
Implicit Generative Model



$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim \delta(x - f_\theta(h))
\end{aligned}
$$

- But what to do with this? What is this?
- $\delta(x - t) = \begin{cases} \infty & x = t \\ 0 & else \end{cases}$.
- $x = f_\theta(h)$.

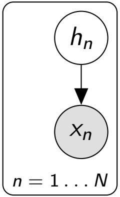
Implicit Generative Model



$$
\begin{aligned}
h &\sim p(h|\theta) \\
x|h &\sim \delta(x - f_\theta(h))
\end{aligned}
$$

- But what to do with this? What is this?
- $\delta(x - t) = \begin{cases} \infty & x = t \\ 0 & else \end{cases}$.
- $x = f_\theta(h)$.
- The usual gig is to marginalize h and maximize the likelihood. (Or equivalently, minimize $KL(p_{data} \| p_{model})$).

Implicit Generative Model



$$\begin{aligned} h &\sim p(h|\theta) \\ x|h &\sim \delta(x - f_\theta(h)) \end{aligned}$$

- But what to do with this? What is this?
- $\delta(x-t) = \begin{cases} \infty & x = t \\ 0 & else \end{cases}.$
- $x = f_\theta(h)$.
- The usual gig is to marginalize h and maximize the likelihood. (Or equivalently, minimize $KL(p_{data} \| p_{model})$).
- Okay, what is p_{model} in this case then?

Here's your answer my friend

[Devroye, Non-Uniform Random Variate Generation, 1986]

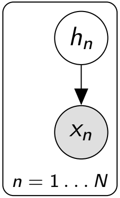
Theorem 4.1.

Let X have distribution function F, and let $h : R \to B$ be a strictly increasing function where B is either R or a proper subset of R. Then $h(X)$ is a random variable with distribution function $F(h^{-1}(x))$.

If F has density f and h^{-1} is absolutely continuous, then $h(X)$ has density

$$(h^{-1})'(x) \ f(h^{-1}(x)), \quad \text{for almost all } x .$$

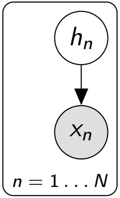
Implicit Generative Model - Toy Example



$$
\begin{aligned}
h &\sim \mathcal{N}(0, \sigma^2) \\
x &= \exp(h)
\end{aligned}
$$

- $f_\theta(h) = \exp(h)$.

Implicit Generative Model - Toy Example

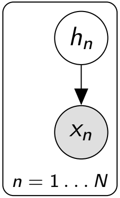


$$\begin{aligned} h &\sim \mathcal{N}(0, \sigma^2) \\ x &= \exp(h) \end{aligned}$$

- $f_\theta(h) = \exp(h)$.
- What is $p(x)$? - I don't know directly, but I know that:

$$\begin{aligned} \Pr(X \leq x) = \Pr(f_\theta(h) \leq x) =& \Pr(h \leq f_\theta^{-1}(x)) \\ \Pr(\exp(h) \leq x) =& \Pr(h \leq \log x), \text{ (for } x \geq 0) \end{aligned}$$

Implicit Generative Model - Toy Example



$$\begin{aligned} h &\sim \mathcal{N}(0, \sigma^2) \\ x &= \exp(h) \end{aligned}$$

- $f_\theta(h) = \exp(h)$.
- What is $p(x)$? - I don't know directly, but I know that:

$$\begin{aligned} \Pr(X \le x) = \Pr(f_\theta(h) \le x) &= \Pr(h \le f_\theta^{-1}(x)) \\ \Pr(\exp(h) \le x) &= \Pr(h \le \log x), \text{ (for } x \ge 0) \end{aligned}$$

- I also know that:

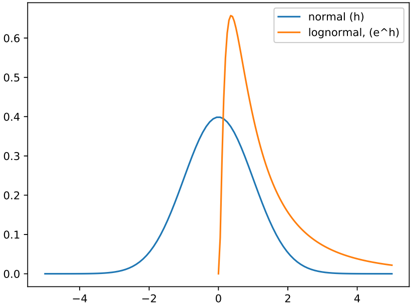
$$p(x) = \frac{\partial}{\partial x} \Pr(h \le \log x) = \frac{\partial}{\partial x} \int_{-\infty}^{\log x} p(h) dh$$

Toy Example - Continued

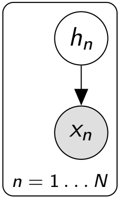
$$
\begin{aligned}
p(x) = \frac{\partial}{\partial x}\Pr(h \leq \log x) =& \frac{\partial}{\partial x}\int_{-\infty}^{\log x} p(h)dh, \text{ for } x \geq 0 \\
=& \frac{1}{x}\mathcal{N}(\log x; 0, \sigma^2) \\
=& \frac{1}{\sqrt{2\pi}\sigma x}\exp\left(\frac{-\log^2 x}{2\sigma^2}\right) \\
=& \mathcal{LN}(0, \sigma^2) \\
\rightarrow& \text{Log-Normal!}
\end{aligned}
$$

Toy Example - Continued

$$
\begin{aligned}
p(x) = \frac{\partial}{\partial x}\Pr(h \le \log x) =& \frac{\partial}{\partial x}\int_{-\infty}^{\log x} p(h)dh, \ \text{for } x \ge 0 \\
=& \frac{1}{x}\mathcal{N}(\log x; 0, \sigma^2) \\
=& \frac{1}{\sqrt{2\pi}\sigma x}\exp\left(\frac{-\log^2 x}{2\sigma^2}\right) \\
=& \mathcal{LN}(0, \sigma^2)
\end{aligned}
$$



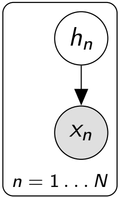
Implicit Generative Model - Toy Example 2



$$
\begin{aligned}
h &\sim \mathcal{N}(0, \sigma^2) \\
x &= h^2
\end{aligned}
$$

- $f_\theta(h) = h^2$.

Implicit Generative Model - Toy Example 2

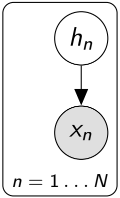


$$h \sim \mathcal{N}(0, \sigma^2)$$
$$x = h^2$$

- $f_\theta(h) = h^2$.
- What is $p(x)$? - I don't know directly, but I know that:

$$\Pr(X \leq x) = \Pr(h \leq f_\theta^{-1}(x))$$

Implicit Generative Model - Toy Example 2



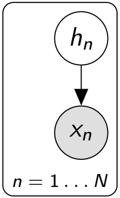
$$h \sim \mathcal{N}(0, \sigma^2)$$
$$x = h^2$$

- $f_\theta(h) = h^2$.
- What is $p(x)$? - I don't know directly, but I know that:

$$\Pr(X \le x) = \Pr(h \le f_\theta^{-1}(x))$$

- Hm. $f_\theta(h)$ is not invertible?

Implicit Generative Model - Toy Example 2



$$h \sim \mathcal{N}(0, \sigma^2)$$
$$x = h^2$$

- $f_\theta(h) = h^2$.
- What is $p(x)$? - I don't know directly, but I know that:

$$\Pr(X \leq x) = \Pr(h \leq f_\theta^{-1}(x))$$

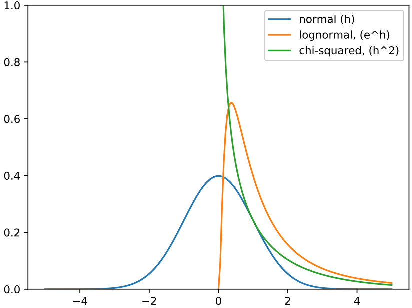
- Hm. $f_\theta(h)$ is not invertible?
- But: $\Pr(h^2 \leq x) = \Pr(|h| \leq \sqrt{x}) = \Pr(h \leq \sqrt{x}) - \Pr(h \leq -\sqrt{x})$, for $x \geq 0$.

Implicit Generative Model - Toy Example 2

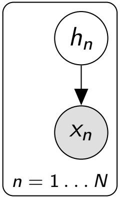
$$
\begin{aligned}
p(x) = \frac{\partial}{\partial x}\Pr(X \leq \log x) =& \frac{\partial}{\partial x}\left(\int_{-\infty}^{\sqrt{x}} p(h)dh - \int_{-\infty}^{-\sqrt{x}} p(h)dh\right),\ x \geq 0 \\
=& \frac{1}{2\sqrt{x}}\left(\mathcal{N}(\sqrt{x}; 0, \sigma^2) + \mathcal{N}(-\sqrt{x}; 0, \sigma^2)\right) \\
=& \frac{1}{\sqrt{2\pi x}\sigma}\left(\exp(-x/2\sigma^2)\right) \\
\rightarrow& \text{Chi-squared distribution.}
\end{aligned}
$$

Implicit Generative Model - Toy Example 2

$$
\begin{aligned}
p(x) = \frac{\partial}{\partial x}\Pr(X \leq \log x) =& \frac{\partial}{\partial x}\left(\int_{-\infty}^{\sqrt{x}} p(h)dh - \int_{-\infty}^{-\sqrt{x}} p(h)dh\right),\ x \geq 0 \\
=& \frac{1}{2\sqrt{x}}\left(\mathcal{N}(\sqrt{x}; 0, \sigma^2) + \mathcal{N}(-\sqrt{x}; 0, \sigma^2)\right) \\
=& \frac{1}{\sqrt{2\pi x}\sigma}\left(\exp(-x/2\sigma^2)\right) \\
\rightarrow& \text{Chi-squared distribution.}
\end{aligned}
$$



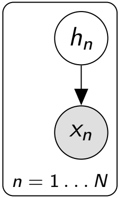
Implicit Generative Model - Real Case



$$h \sim \mathcal{N}(0, \sigma^2)$$
$$x = f_\theta(h)$$

- $f_\theta(h)$ is an arbitrary function now. Let's consider a one dimensional neural net, such that $f_\theta(h) = \sigma(\theta h)$, where $\sigma(.)$ is some typical neural net non-linearity.

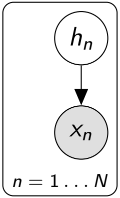
Implicit Generative Model - Real Case



$$
\begin{aligned}
h &\sim \mathcal{N}(0, \sigma^2) \\
x &= f_\theta(h)
\end{aligned}
$$

- $f_\theta(h)$ is an arbitrary function now. Let's consider a one dimensional neural net, such that $f_\theta(h) = \sigma(\theta h)$, where $\sigma(.)$ is some typical neural net non-linearity.
- Can we analytically derive $p(x)$ now? Maybe. But let's consider what we need to do in the general case.

Implicit Generative Model - Real Case

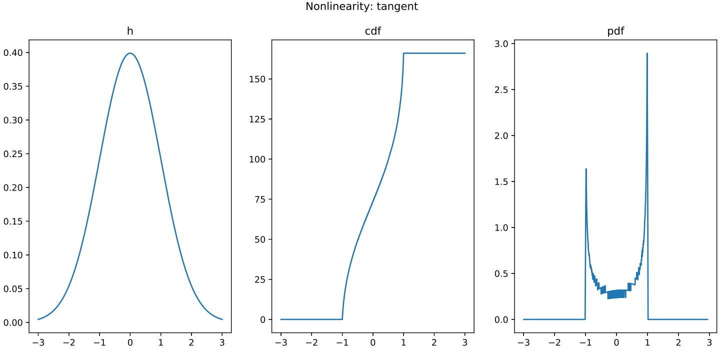


$$h \sim \mathcal{N}(0, \sigma^2)$$
$$x = f_\theta(h)$$

- $f_\theta(h)$ is an arbitrary function now. Let's consider a one dimensional neural net, such that $f_\theta(h) = \sigma(\theta h)$, where $\sigma(.)$ is some typical neural net non-linearity.
- Can we analytically derive $p(x)$ now? Maybe. But let's consider what we need to do in the general case.
- $p(x) = \frac{\partial}{\partial x} \int_{f_\theta(h) \leq x} p(h) dh$. $\rightarrow$ for all $x \in \mathbb{R}$, we need to find the set $\{h : f(h) \leq x, h \in \mathbb{R}\}$. In 1-D, we can hope to do something numerically.

Visualizing output densities

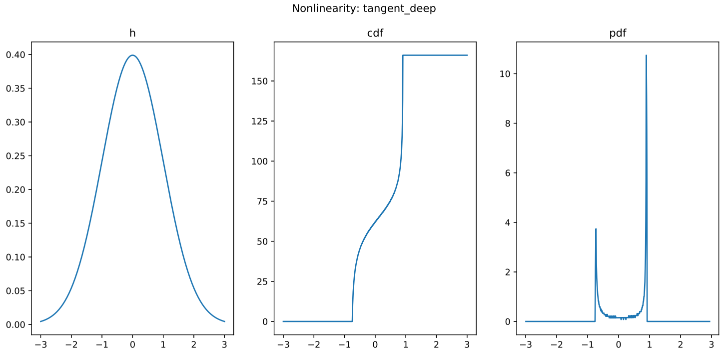
$$f_\theta = \tanh(1.4h + 0.2)$$



(Jaggedness is due to numerical issues)

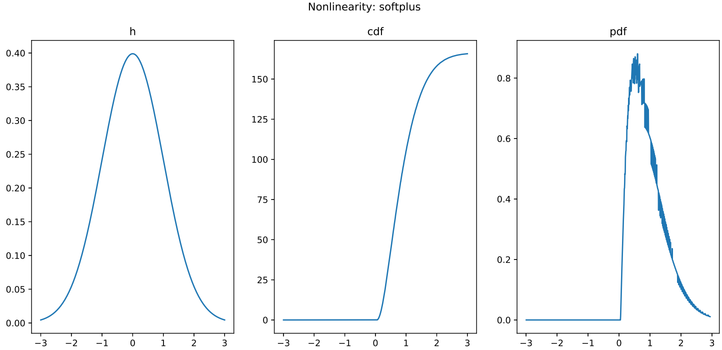
Visualizing output densities

$$f_\theta = \tanh(1.4\tanh(1.4\tanh(1.4h + 0.2) + 0,2) + 0.2)$$



Visualizing output densities

$$f_\theta = \log(\exp(h + 0.2) + 1)$$



Multidimensional Case

- In the multidimensional case, we need to compute the set $\mathcal{S}(x) := \{h : f_\theta(h) \leq x, h \in \mathbb{R}^K, x \in \mathbb{R}^L\}$. To compute the multi-dimensional pdf:

$$p(x) = \frac{\partial}{\partial x} \int_{h \in S(x)} p(h) dh$$

Multidimensional Case

- In the multidimensional case, we need to compute the set $\mathcal{S}(x) := \{h : f_\theta(h) \leq x, h \in \mathbb{R}^K, x \in \mathbb{R}^L\}$. To compute the multi-dimensional pdf:

$$p(x) = \frac{\partial}{\partial x} \int_{h \in S(x)} p(h) dh$$

- We cannot explicitly compute this set in practice.

Multidimensional Case

- In the multidimensional case, we need to compute the set $\mathcal{S}(x) := \{h : f_\theta(h) \leq x, h \in \mathbb{R}^K, x \in \mathbb{R}^L\}$. To compute the multi-dimensional pdf:

$$p(x) = \frac{\partial}{\partial x} \int_{h \in S(x)} p(h) dh$$

- We cannot explicitly compute this set in practice.
- But we still need an handle on $p_{model}(.)$ to train our forward mapping $f_\theta(.)$.

Multidimensional Case

- In the multidimensional case, we need to compute the set $\mathcal{S}(x) := \{h : f_\theta(h) \leq x, h \in \mathbb{R}^K, x \in \mathbb{R}^L\}$. To compute the multi-dimensional pdf:

$$p(x) = \frac{\partial}{\partial x} \int_{h \in S(x)} p(h) dh$$

- We cannot explicitly compute this set in practice.
- But we still need an handle on $p_{model}(.)$ to train our forward mapping $f_\theta(.)$.
- **Good news:** It is very easy to sample from implicit generative models!

Plan

Plan

Training the forward model by moment-matching:

- We can match the expected output moment with data moments:

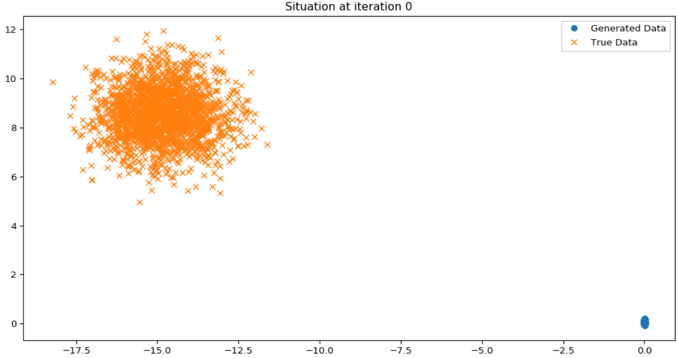
$$\min_{\theta} \|\mathbb{E}_{p(h)}[s(f_\theta(h))] - \mathbb{E}_{p_{\text{data}}(x_{\text{data}})}[s(x_{\text{data}})]\|_2^2,$$

$$\approx \min_{\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} s(f_\theta(h_n)) - \frac{1}{N} \sum_{n'=1}^{N} s(x_{n'}^{\text{data}}) \right\|_2^2$$

where $s(.)$ is some summary statistics (e.g. covariance).

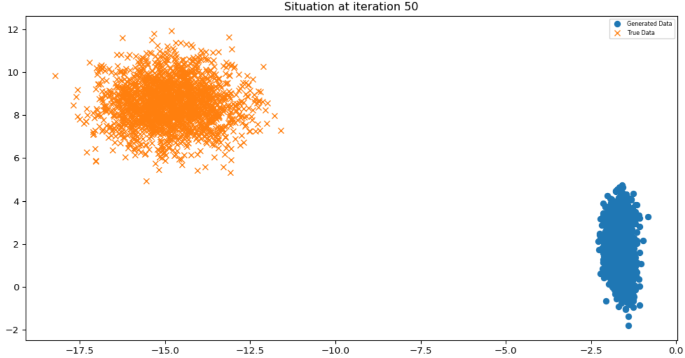
Moment Matching in Action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$



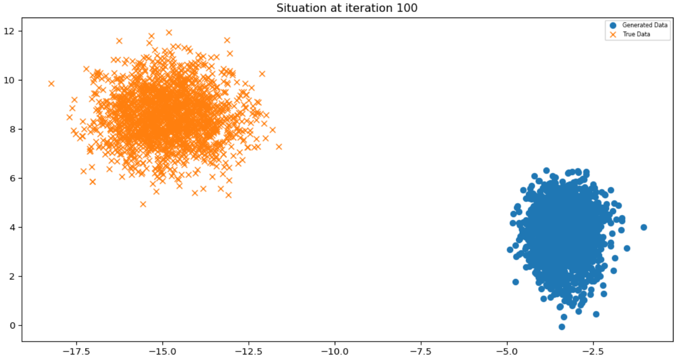
Moment Matching in Action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$



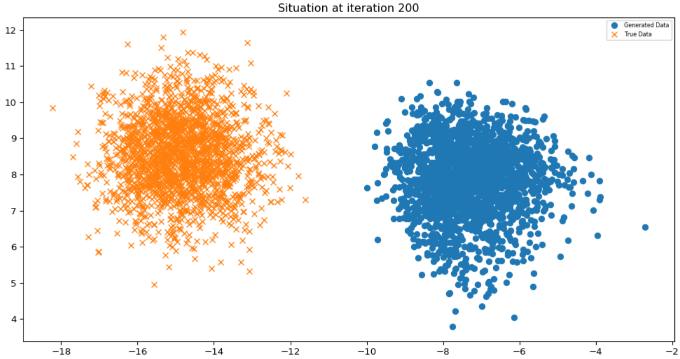
Moment Matching in Action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$



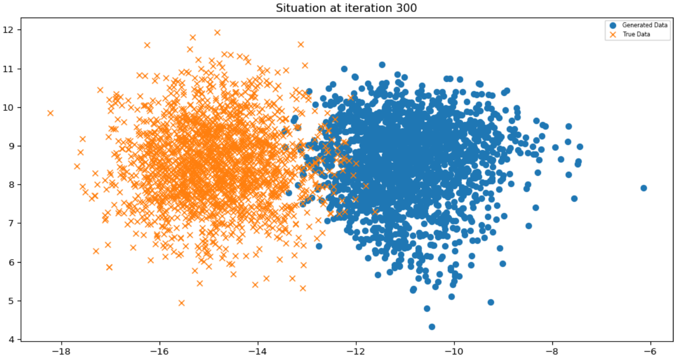
Moment Matching in Action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$



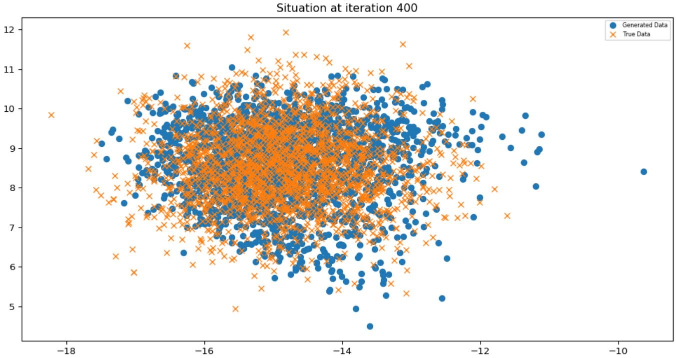
Moment Matching in Action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$



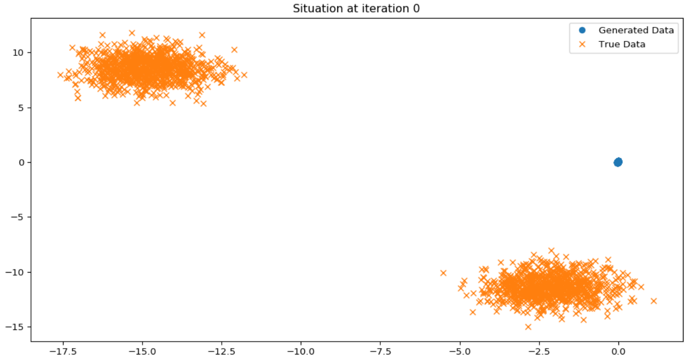
Moment Matching in Action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$

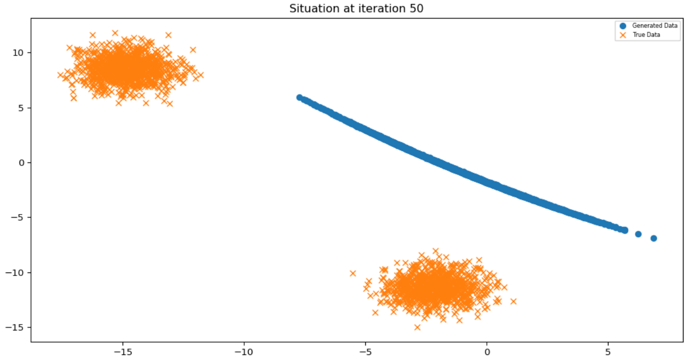


Seems to work fine in this toy case.
But what would happen with slightly more difficult data?

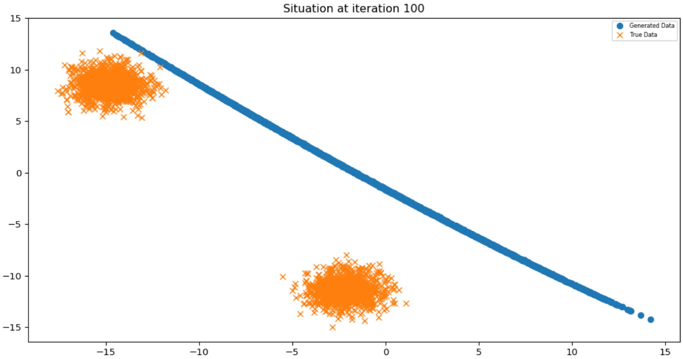
Moment Matching in action, case 2



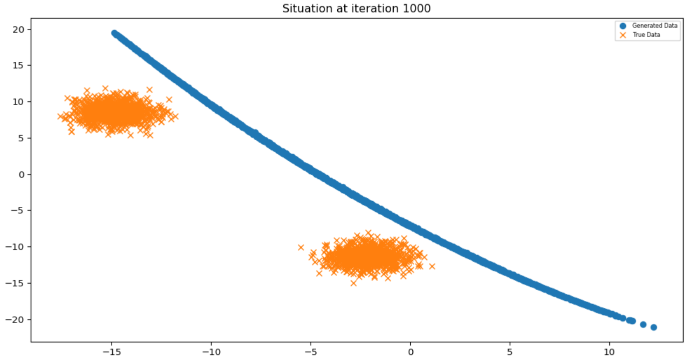
Moment Matching in action, case 2



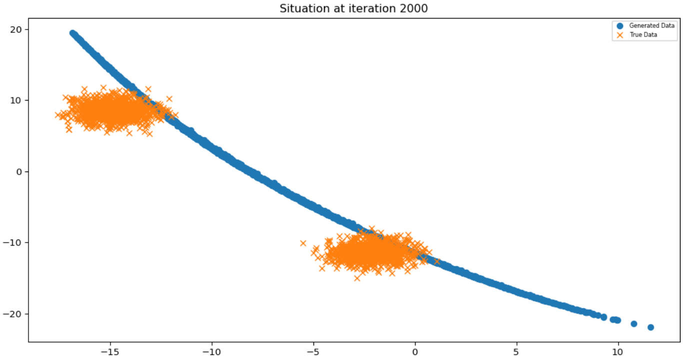
Moment Matching in action, case 2



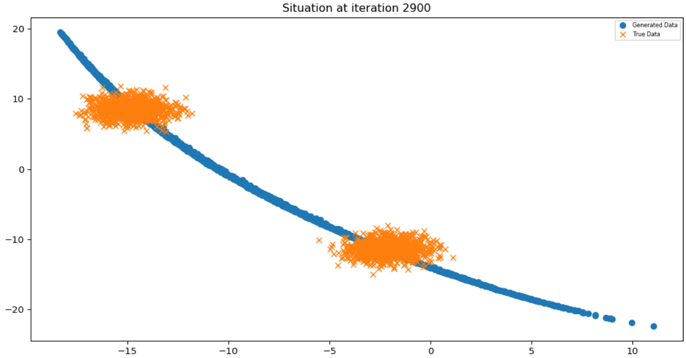
Moment Matching in action, case 2



Moment Matching in action, case 2

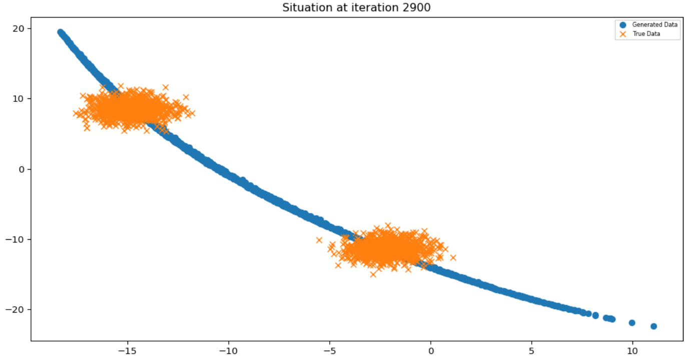


Moment Matching in action, case 2



Horrible, but expected.

Moment Matching in action, case 2

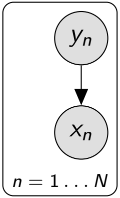


Horrible, but expected.
Choice of sufficient statistics is crucial - this is against the point.
Can we do something more agnostic?

Plan

Ratio Estimation

Now let's consider this mixture model:



$$\begin{aligned} y &\sim \mathcal{BE}(\pi) \\ x|y &\sim p_{model}(x)^{[y=0]} p_{data}(x)^{[y=1]} \end{aligned}$$

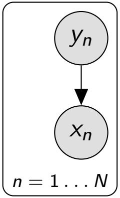
- $y = 0$, means generated from the model, $y = 1$ means the item is from the dataset. Write the joint distribution:

$$p(x, y) = (\pi p_{model}(x))^{[y=0]} \left((1 - \pi) p_{data}(x)\right)^{[y=1]}$$

Then what are the class posteriors $p(y = 0|x)$, and $p(y = 1|x)$?

Ratio Estimation

Now let's consider this mixture model:



$$
\begin{aligned}
y &\sim \mathcal{BE}(\pi) \\
x|y &\sim p_{model}(x)^{[y=0]} p_{data}(x)^{[y=1]}
\end{aligned}
$$

- $y = 0$, means generated from the model, $y = 1$ means the item is from the dataset. Write the joint distribution:

$$p(x, y) = (\pi p_{model}(x))^{[y=0]} \left((1 - \pi) p_{data}(x)\right)^{[y=1]}$$

  Then what are the class posteriors $p(y = 0|x)$, and $p(y = 1|x)$?
- Apply Bayes' rule:

$$p(y|x) = \frac{(\pi p_{model}(x))^{[y=0]} \left((1 - \pi) p_{data}\right)^{[y=1]}}{\pi p_{model}(x) + (1 - \pi) p_{data}(x)}$$

Ratio Estimation - continued

- Now let's write down the log likelihood for the posterior over y (and assume $\pi = 0.5$, which you don't have to but original paper does):

$$\log p(y_{1:N}|x) = \sum_n [y_n = 1] \log r(x_n) + [y_n = 0] \log 1 - r(x_n),$$

where $r(x) := \frac{p_{data}(x)}{p_{data}(x)+p_{model}(x)}$.

Ratio Estimation - continued

- Now let's write down the log likelihood for the posterior over y (and assume $\pi = 0.5$, which you don't have to but original paper does):

$$\log p(y_{1:N}|x) = \sum_n [y_n = 1] \log r(x_n) + [y_n = 0] \log 1 - r(x_n),$$

where $r(x) := \frac{p_{data}(x)}{p_{data}(x)+p_{model}(x)}$.

- But, we do not know these densities, do we? Whatever, let's try to "learn" $r(x)$ from data. Let's replace it with a parametric binary classifier $D_\xi(x)$, and call $\log p(y|x)$, $\mathcal{L}(\xi, \theta)$:

$$\mathcal{L}(\xi, \theta) = \sum_n [y_n = 1] \log D_\xi(x_n) + [y_n = 0] \log 1 - D_\xi(x_n),$$

Ratio Estimation - continued

- Now let's write down the log likelihood for the posterior over y (and assume $\pi = 0.5$, which you don't have to but original paper does):

$$\log p(y_{1:N}|x) = \sum_n [y_n = 1] \log r(x_n) + [y_n = 0] \log 1 - r(x_n),$$

where $r(x) := \frac{p_{data}(x)}{p_{data}(x) + p_{model}(x)}$.

- But, we do not know these densities, do we? Whatever, let's try to "learn" $r(x)$ from data. Let's replace it with a parametric binary classifier $D_\xi(x)$, and call $\log p(y|x)$, $\mathcal{L}(\xi, \theta)$:

$$\mathcal{L}(\xi, \theta) = \sum_n [y_n = 1] \log D_\xi(x_n) + [y_n = 0] \log 1 - D_\xi(x_n),$$

- Also get rid of the model/data indicators $y_{1:N}$ using our implicit generative model:

$$\mathcal{L}(\xi, \theta) = \sum_{n:[y_n=1]} \log D_\xi(x_n) + \sum_{n:[y_n=0]} \log 1 - D_\xi(f_\theta(h_n)),$$

- Now, maximize with respect to ξ to approximate $r(x)$ as best as possible. Minimize with respect θ to maximize $p_{model}/(p_{model} + p_{data})$.

Ratio Estimation - continued

- Now, maximize with respect to ξ to approximate $r(x)$ as best as possible. Minimize with respect θ to maximize $p_{model}/(p_{model} + p_{data})$.

$$\min_{\theta} \max_{\xi} \mathcal{L}(\xi, \theta) = \min_{\theta} \max_{\xi} \sum_{n} \log D_{\xi}(x_n) + \sum_{n} \log 1 - D_{\xi}(f_{\theta}(h_n)),$$

- Here's your glorified Generative Adversarial Network! In practice you do: (actually don't) 5 iterations of:

$$\max_{\xi} \sum_{n} \log D_{\xi}(x_n) + \sum_{n} \log 1 - D_{\xi}(f_{\theta}(h_n)),$$

  Then, flip the signs and do:

$$\max_{\theta} \sum_{n} \log D_{\xi}(f_{\theta}(h_n)),$$

A “By the way” slide:

- The 'best' strategy:

$$\int \log D(x) p_{data}(x) dx + \int \log(1 - D(x)) p_{model}(x) dx$$
$$\approx \sum_n [y_n = 1] \log D_\xi(x_n) + \sum_n [y_n = 0] \log 1 - D_\xi(x_n),$$

A “By the way” slide:

- The 'best' strategy:

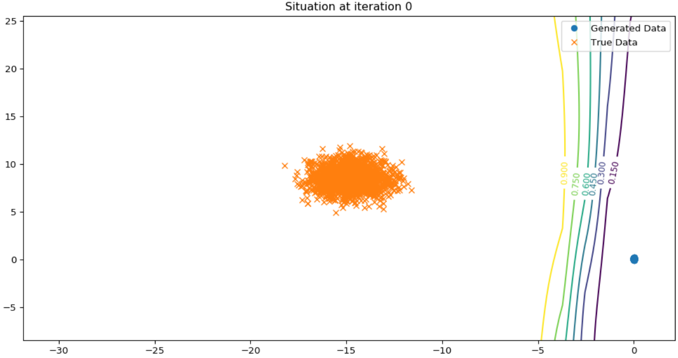
$$\int \log D(x) p_{data}(x) dx + \int \log(1 - D(x)) p_{model}(x) dx$$
$$\approx \sum_n [y_n = 1] \log D_\xi(x_n) + \sum_n [y_n = 0] \log 1 - D_\xi(x_n),$$

- Therefore the optimal classifier $D(x)$ is:

$$\frac{\partial}{\partial D(x)} \left(\int \log D(x) p_{data}(x) dx + \int \log(1 - D(x)) p_{model}(x) dx \right) = 0$$
$$\rightarrow \frac{p_{data}(x)}{D(x)} - \frac{p_{model}(x)}{1 - D(x)} = 0$$
$$\rightarrow D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{model}(x)}$$

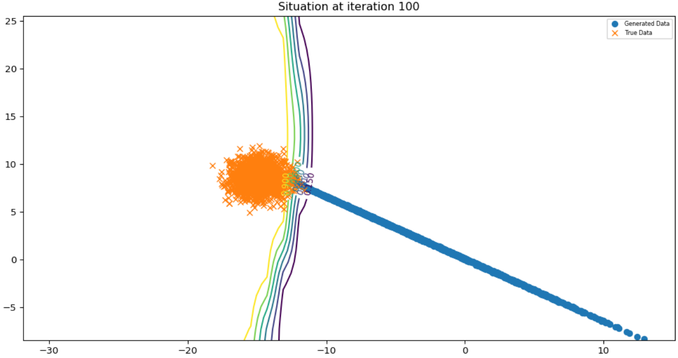
Let's see some GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



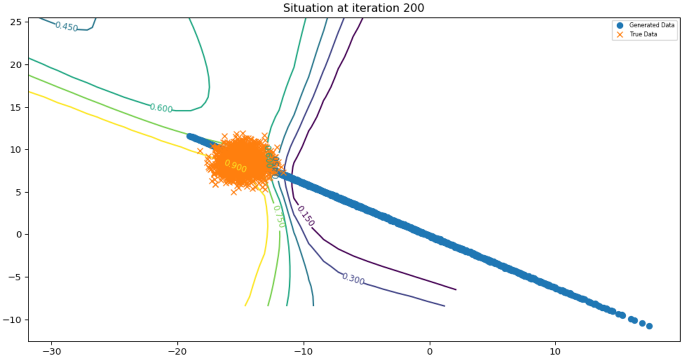
Let's see some GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



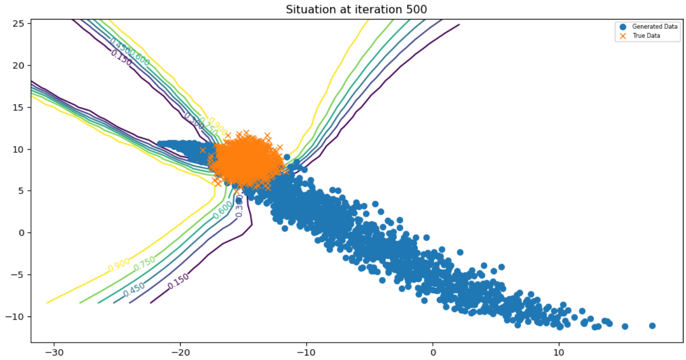
Let's see some GAN action

$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$ (Same forward mapping as before)



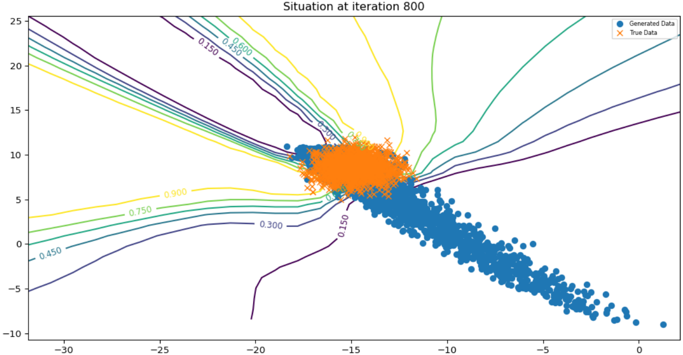
Let's see some GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



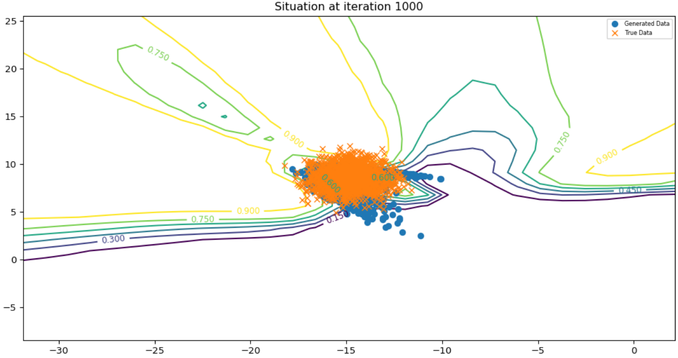
Let's see some GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



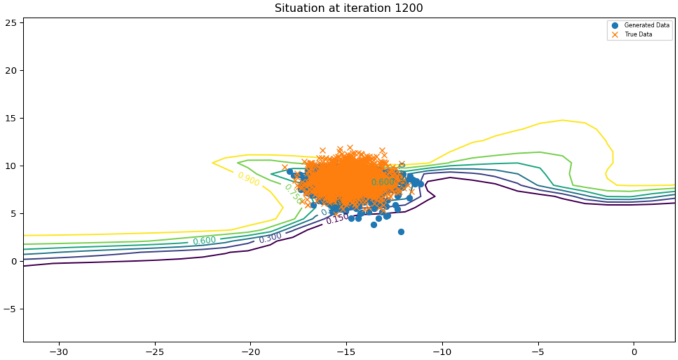
Let's see some GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



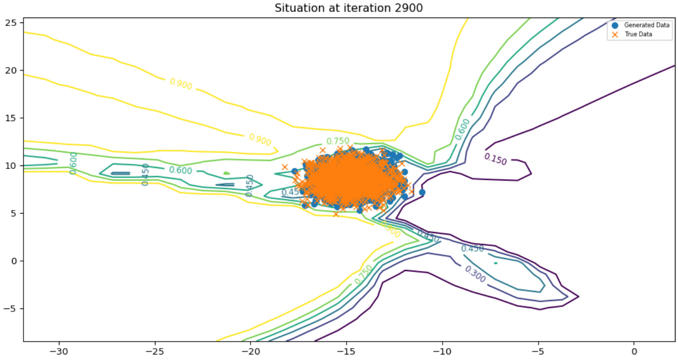
Let's see some GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



Let's see some GAN action

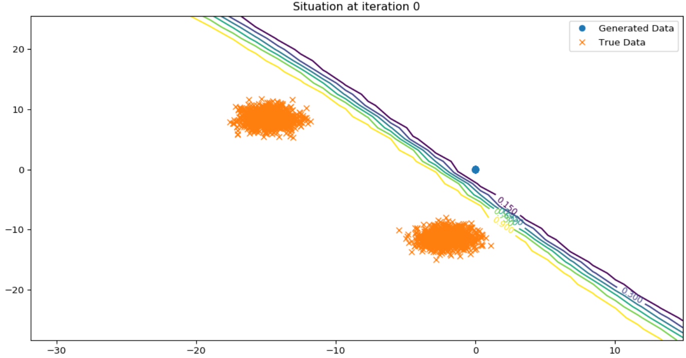
$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



Seems to work fine in this toy case.
But what would happen in our good old mixture example?

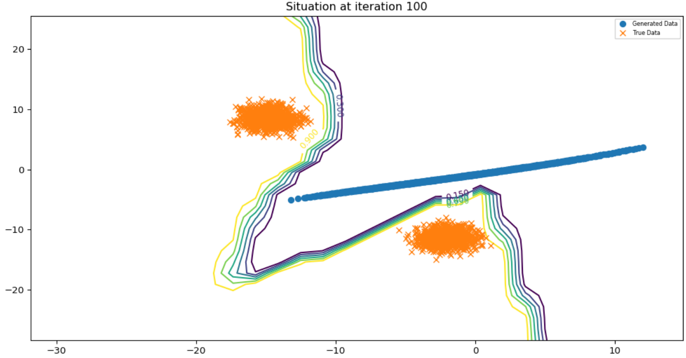
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



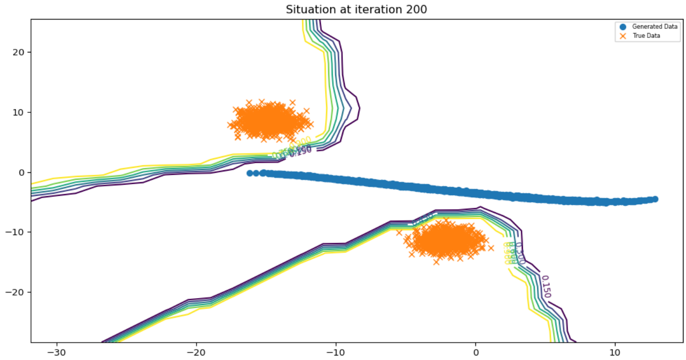
More GAN action

$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$ (Same forward mapping as before)



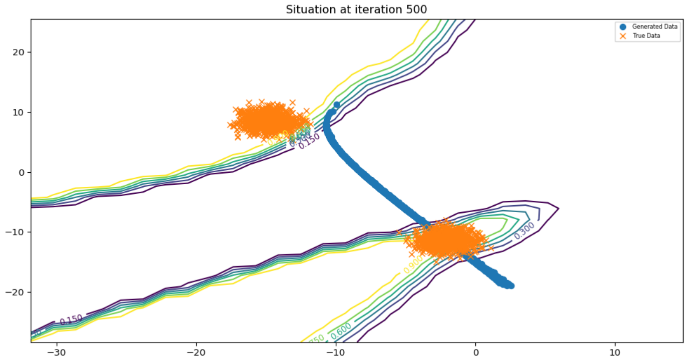
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



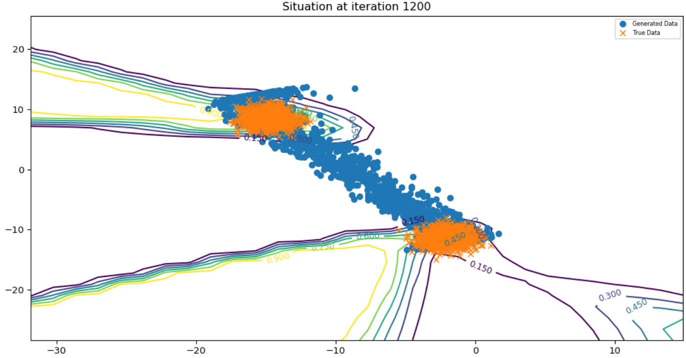
More GAN action

$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$ (Same forward mapping as before)



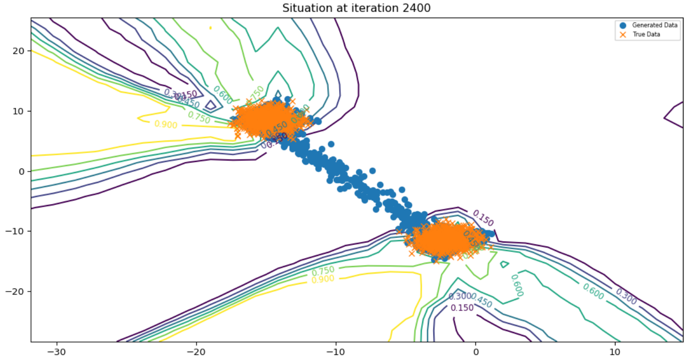
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



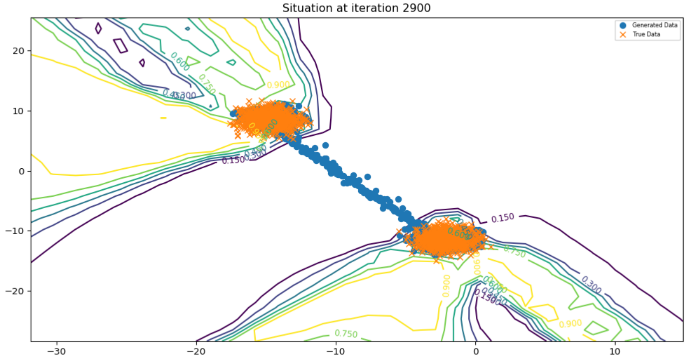
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$$ (Same forward mapping as before)



More GAN action

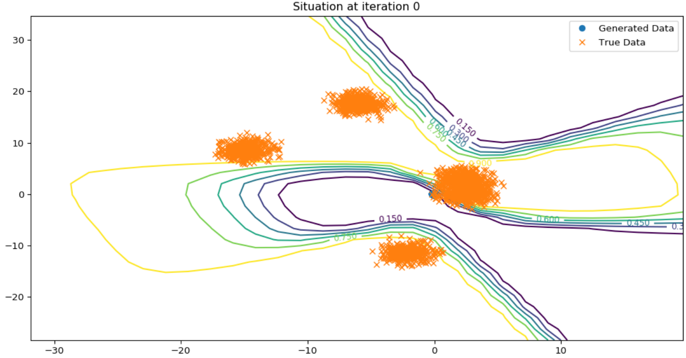
$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



Sometimes we see this "smearing" behavior.

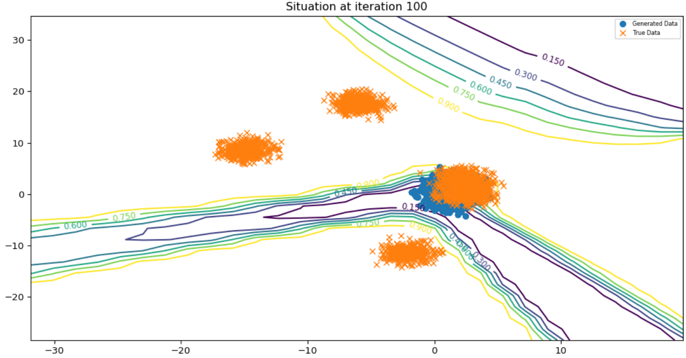
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



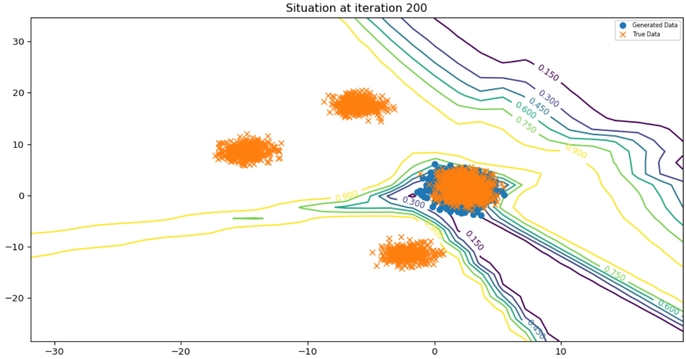
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



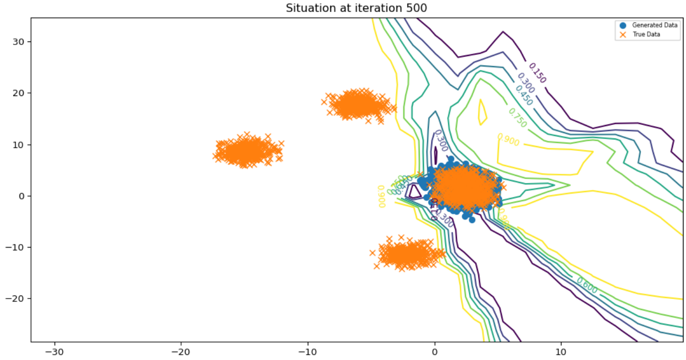
More GAN action

$$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2 \text{ (Same forward mapping as before)}$$



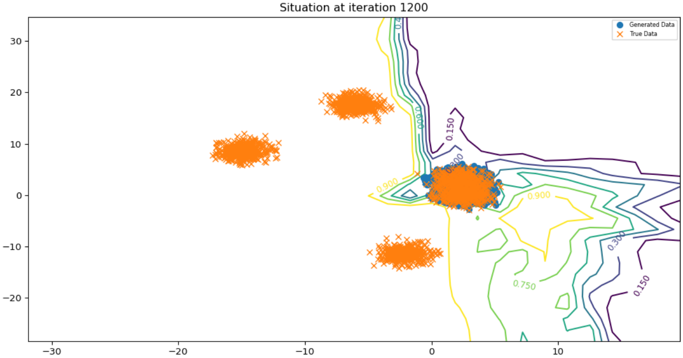
More GAN action

$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$ (Same forward mapping as before)



More GAN action

$f_\theta(h) = W_2 \tanh(W_1 h + b_1) + b_2$ (Same forward mapping as before)



Sometimes generator "collapses onto" a subset of the modes.

Little bit on collapsing

- Mode collapse is a big issue.
- Wasserstein-Gans (which approximately minimizes Wasserstein-1 distance between $p_{model}(x)$, and $p_{data}(x)$. This results in smoother gradients.
- Bayesian GANs [Saatci, 2017], integrates out θ and ξ. Claim is that the additional work pays off very well.

Conclusions:

- We looked at the implicit generative models.
- GANs are a special case of implicit generative model learning.
- Things I couldn't discuss: Wasserstein GANs, f-GANs.