Variational Autoencoders

Presented by Alex Beatson

Materials from Yann LeCun, Jaan Altosaar, Shakir Mohamed

Contents

1. Why unsupervised learning, and why generative models?
   (Selected slides from Yann LeCun's keynote at NIPS 2016)
2. What is a variational autoencoder?
   (Jaan Altosaar's blog post)
3. A simple derivation of the VAE objective from importance sampling
   (Shakir Mohamed's slides)

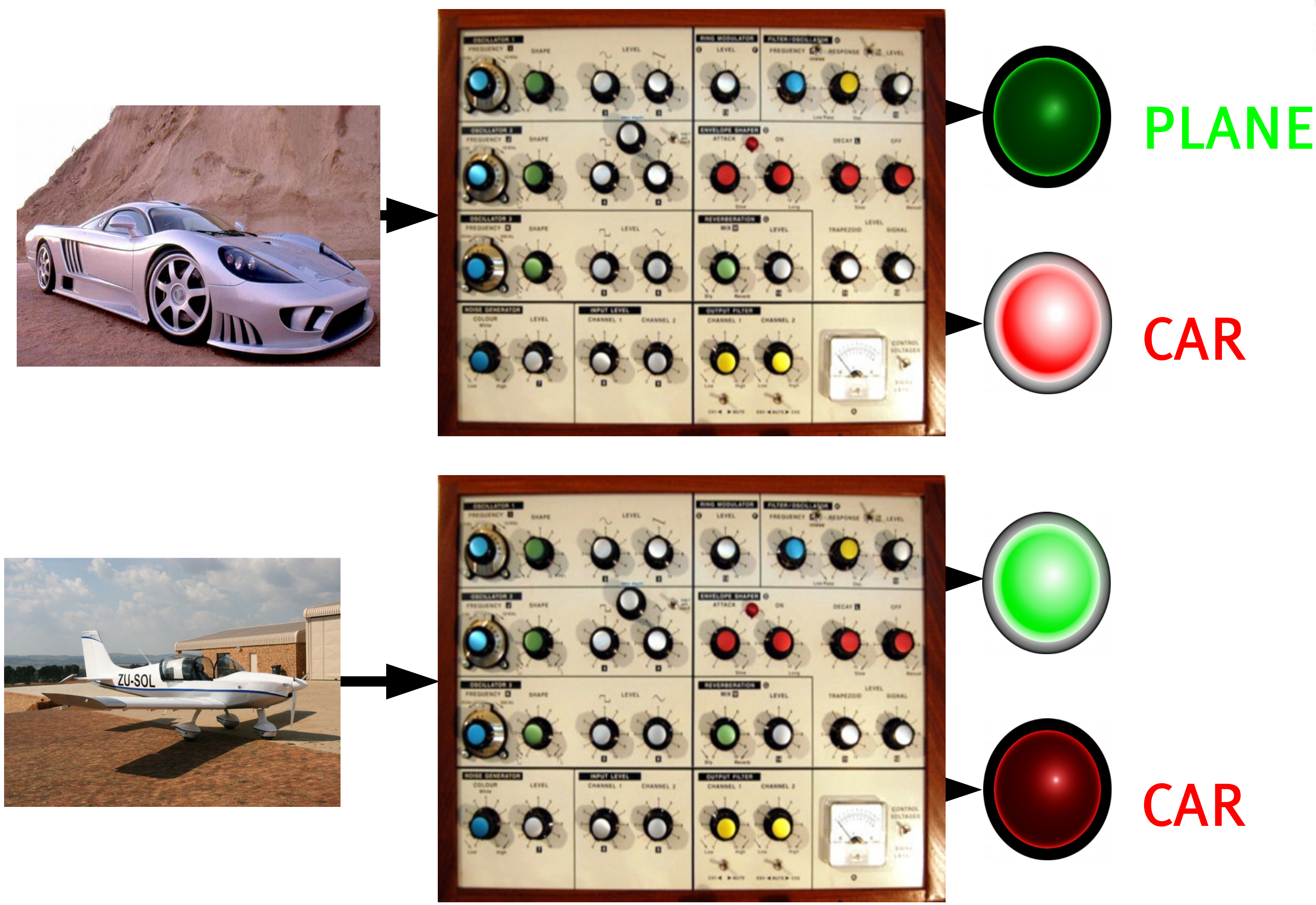
   Sections 2 and 3 were done as a chalk talk in the presentation

1. Why unsupervised learning, and why generative models?

- Selected slides from Yann LeCun's keynote at NIPS 2016

Supervised Learning

- We can train a machine on lots of examples of tables, chairs, dog, cars, and people
- But will it recognize table, chairs, dogs, cars, and people it has never seen before?



Obstacles to Progress in AI

- **Machines need to learn/understand how the world works**
  - Physical world, digital world, people,….
  - They need to acquire some level of common sense
- **They need to learn a very large amount of background knowledge**
  - Through observation and action
- **Machines need to perceive the state of the world**
  - So as to make accurate predictions and planning
- **Machines need to update and remember estimates of the state of the world**
  - Paying attention to important events. Remember relevant events
- **Machines neet to reason and plan**
  - Predict which sequence of actions will lead to a desired state of the world

- **Intelligence & Common Sense =**

  Perception + Predictive Model + Memory + Reasoning & Planning

What is Common Sense?

- "The trophy doesn't fit in the suitcase because it's too large/small"
  - (winograd schema)

- "Tom picked up his bag and left the room"

- We have common sense because we know how the world works

- How do we get machines to learn that?

Common Sense is the ability to fill in the blanks

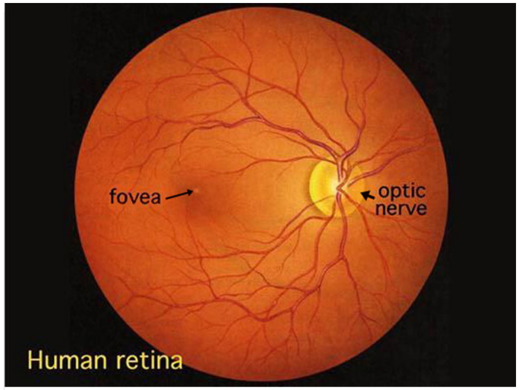


Fig. 1. Human retina as seen through an opthalmoscope.

- Infer the state of the world from partial information
- Infer the future from the past and present
- Infer past events from the present state

- Filling in the visual field at the retinal blind spot
- Filling in occluded images
- Fillling in missing segments in text, missing words in speech.
- Predicting the consequences of our actions
- Predicting the sequence of actions leading to a result

- Predicting any part of the past, present or future percepts from whatever information is available.

- That’s what predictive learning is
- But really, that’s what many people mean by unsupervised learning

The Necessity of Unsupervised Learning / Predictive Learning

- The number of samples required to train a large learning machine (for any task) depends on the amount of information that we ask it to predict.
  - The more you ask of the machine, the larger it can be.

- “The brain has about 10^14 synapses and we only live for about 10^9 seconds. So we have a lot more parameters than data. This motivates the idea that we must do a lot of unsupervised learning since the perceptual input (including proprioception) is the only place we can get 10^5 dimensions of constraint per second.”
  - Geoffrey Hinton (in his 2014 AMA on Reddit)
  - (but he has been saying that since the late 1970s)

- Predicting human-provided labels is not enough

- Predicting a value function is not enough

How Much Information Does the Machine Need to Predict?

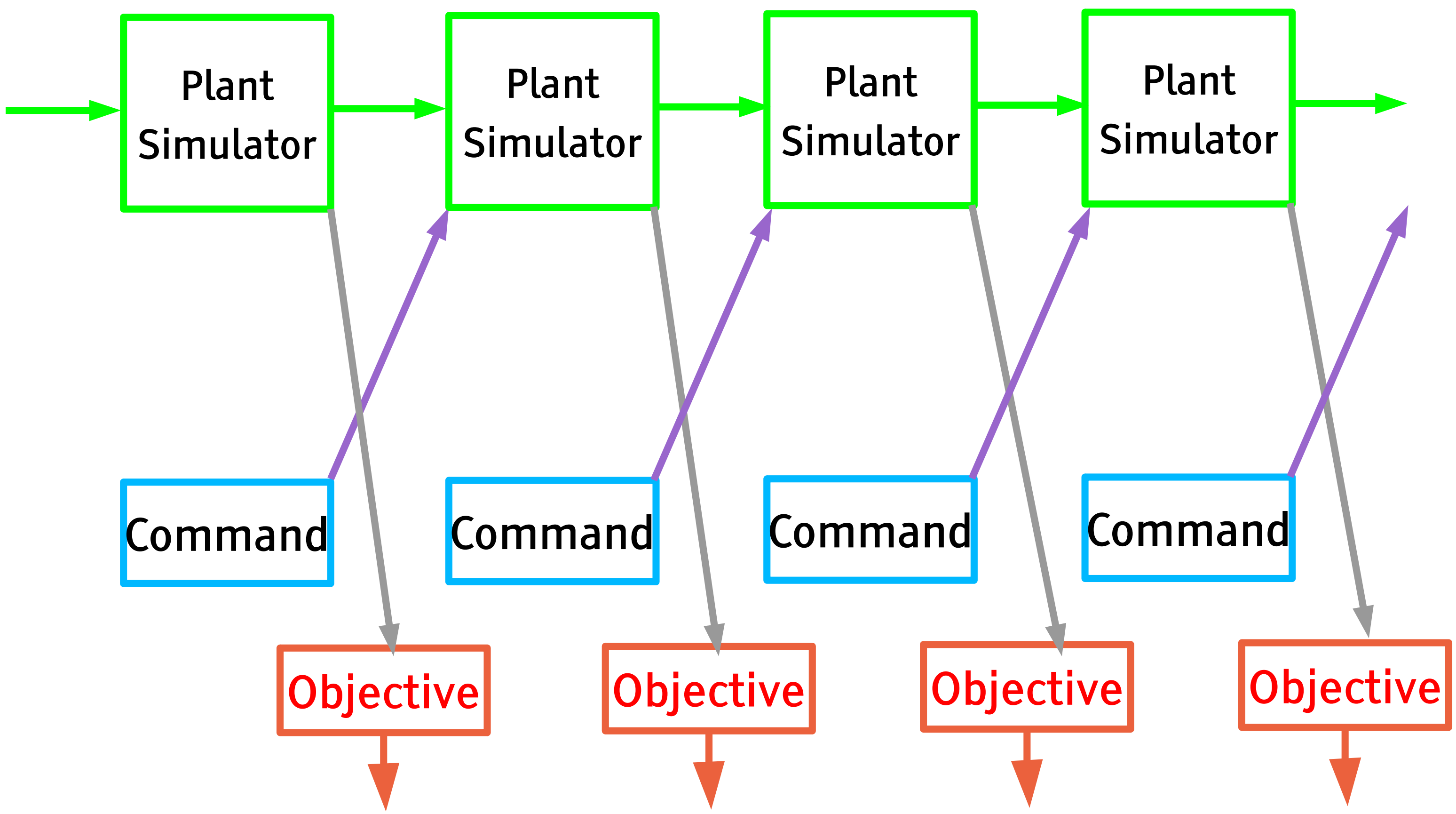
- "Pure" Reinforcement Learning (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**

- Supervised Learning (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- Unsupervised/Predictive Learning (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Classical model-based optimal control

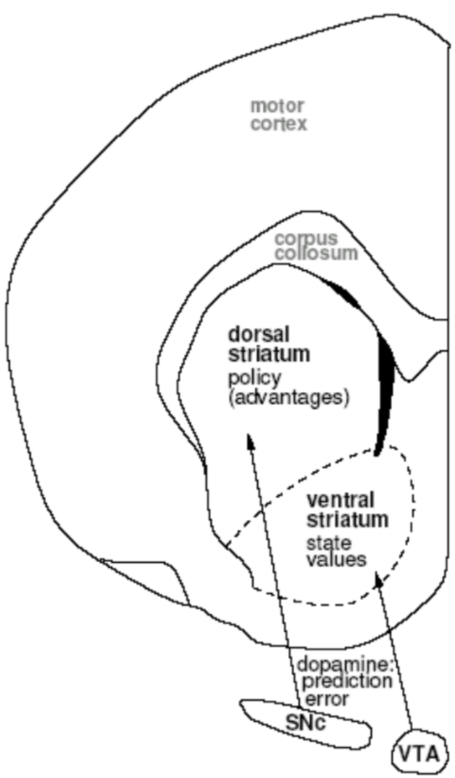
- **Simulate the world (the plant) with an initial control sequence**
- **Adjust the control sequence to optimize the objective through gradient descent**
- **Backprop through time was invented by control theorists in the late 1950s**
  - it's sometimes called the adjoint state method
  - [Athans & Falb 1966, Bryson & Ho 1969]

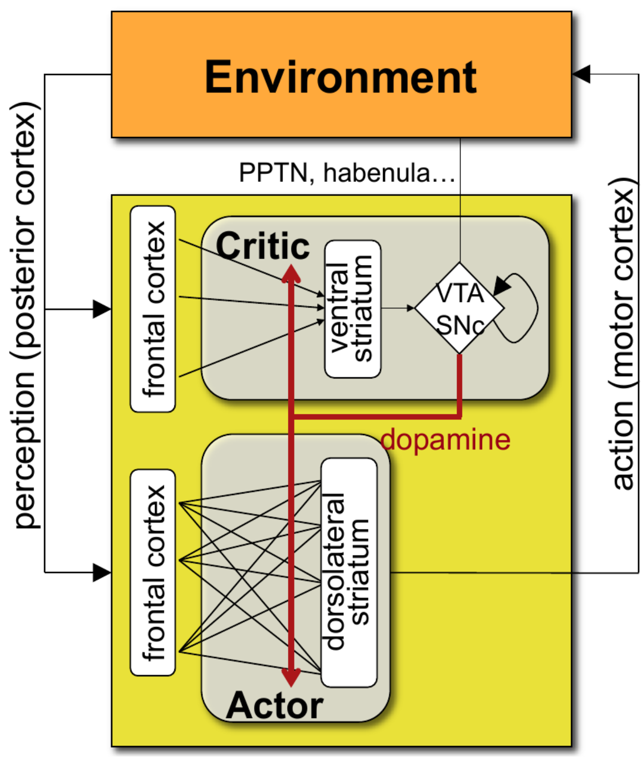


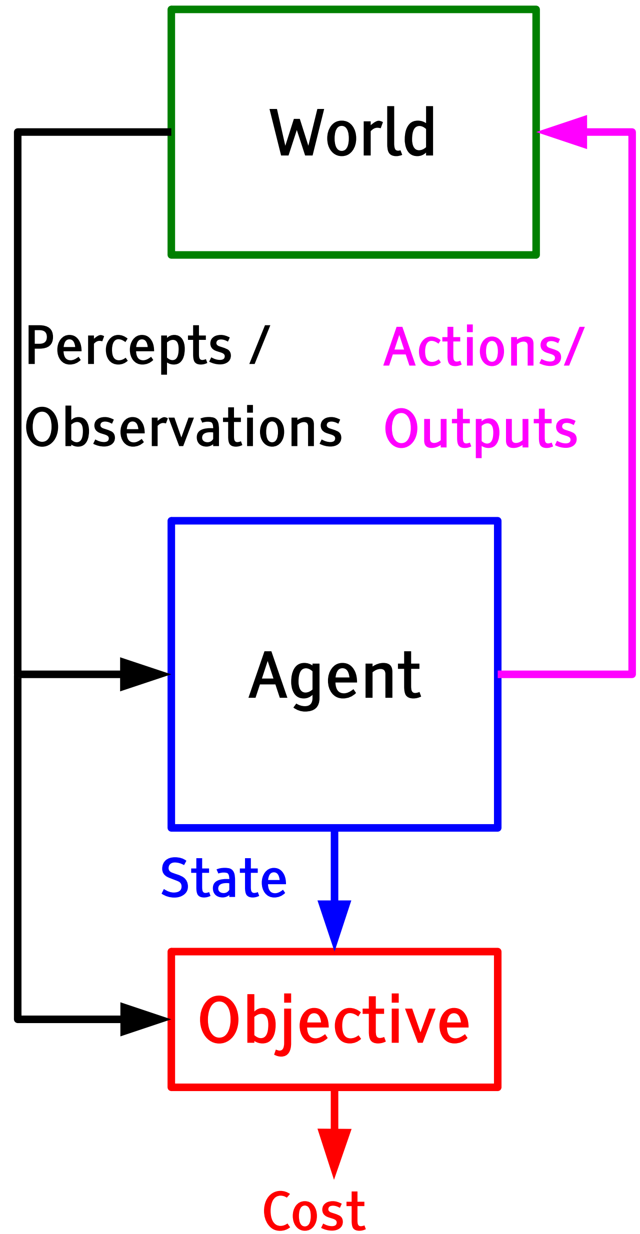
The Architecture
Of an
Intelligent System

AI System: Learning Agent + Immutable Objective

- The agent gets percepts from the world
- The agent acts on the world
- The agents tries to minimize the long-term expected cost.

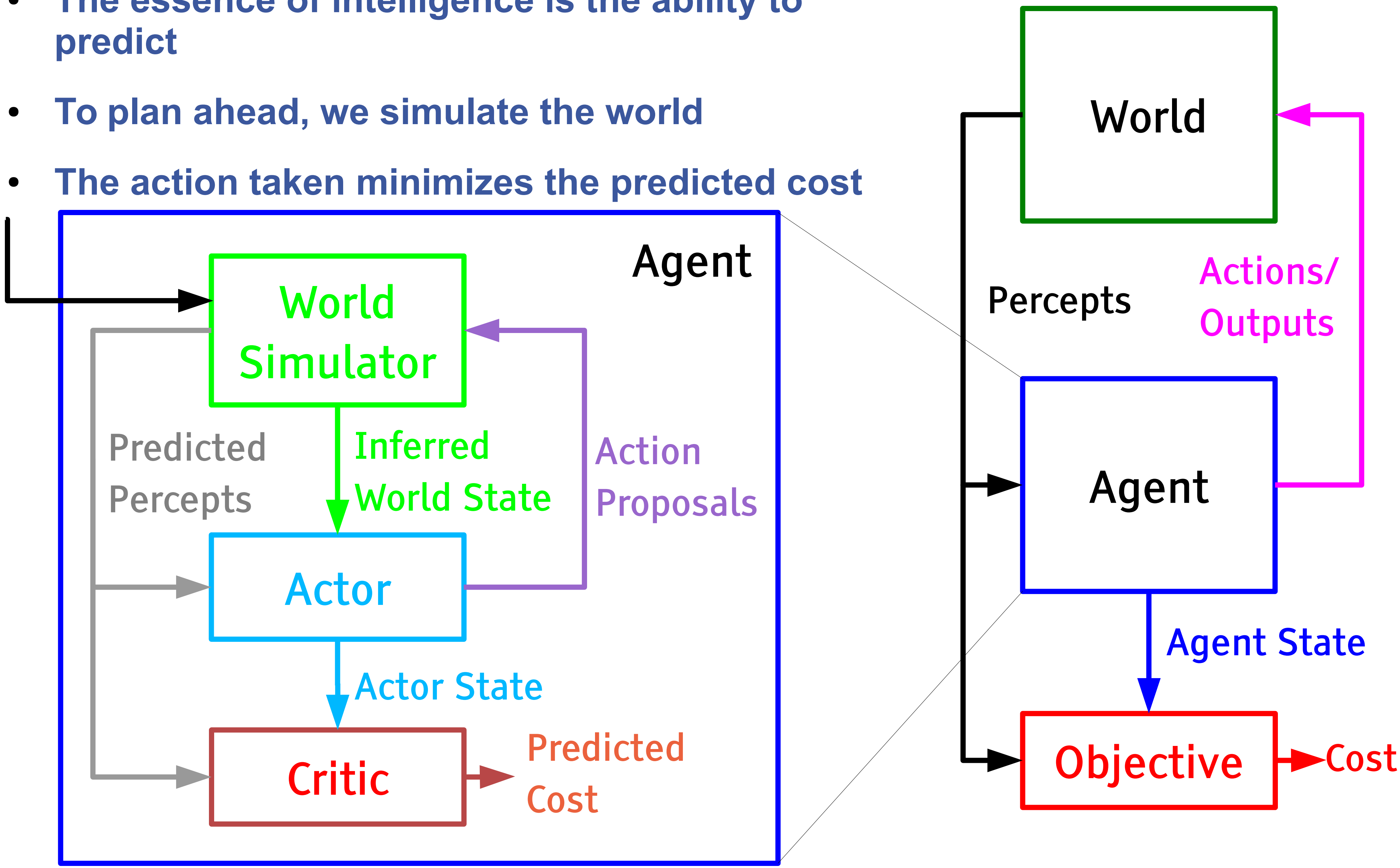






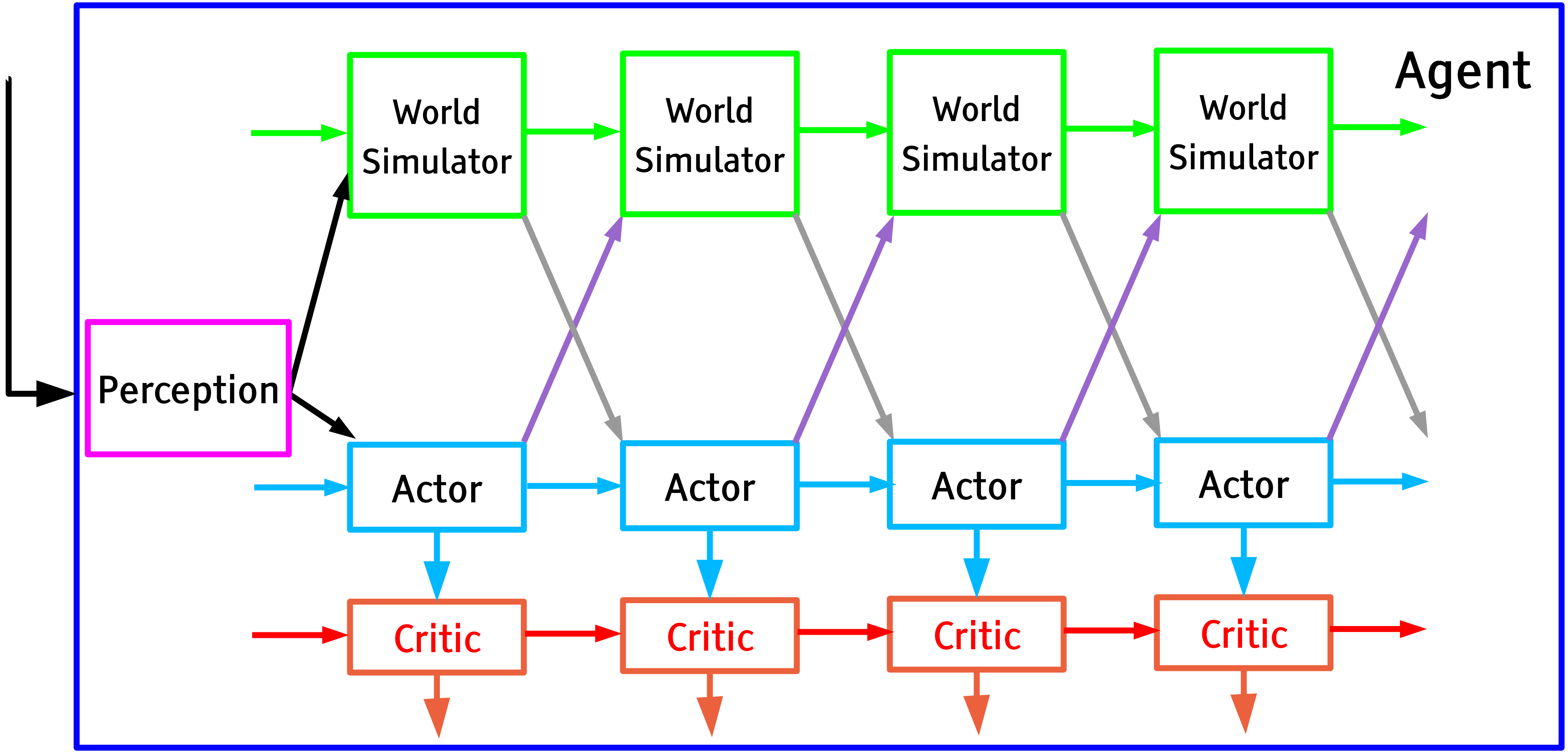
AI System: Predicting + Planning = Reasoning

- The essence of intelligence is the ability to predict
- To plan ahead, we simulate the world
- The action taken minimizes the predicted cost



What we need is Model-Based Reinforcement Learning

- The essence of intelligence is the ability to predict
- To plan ahead, we must simulate the world, so as to minimizes the predicted value of some objective function.

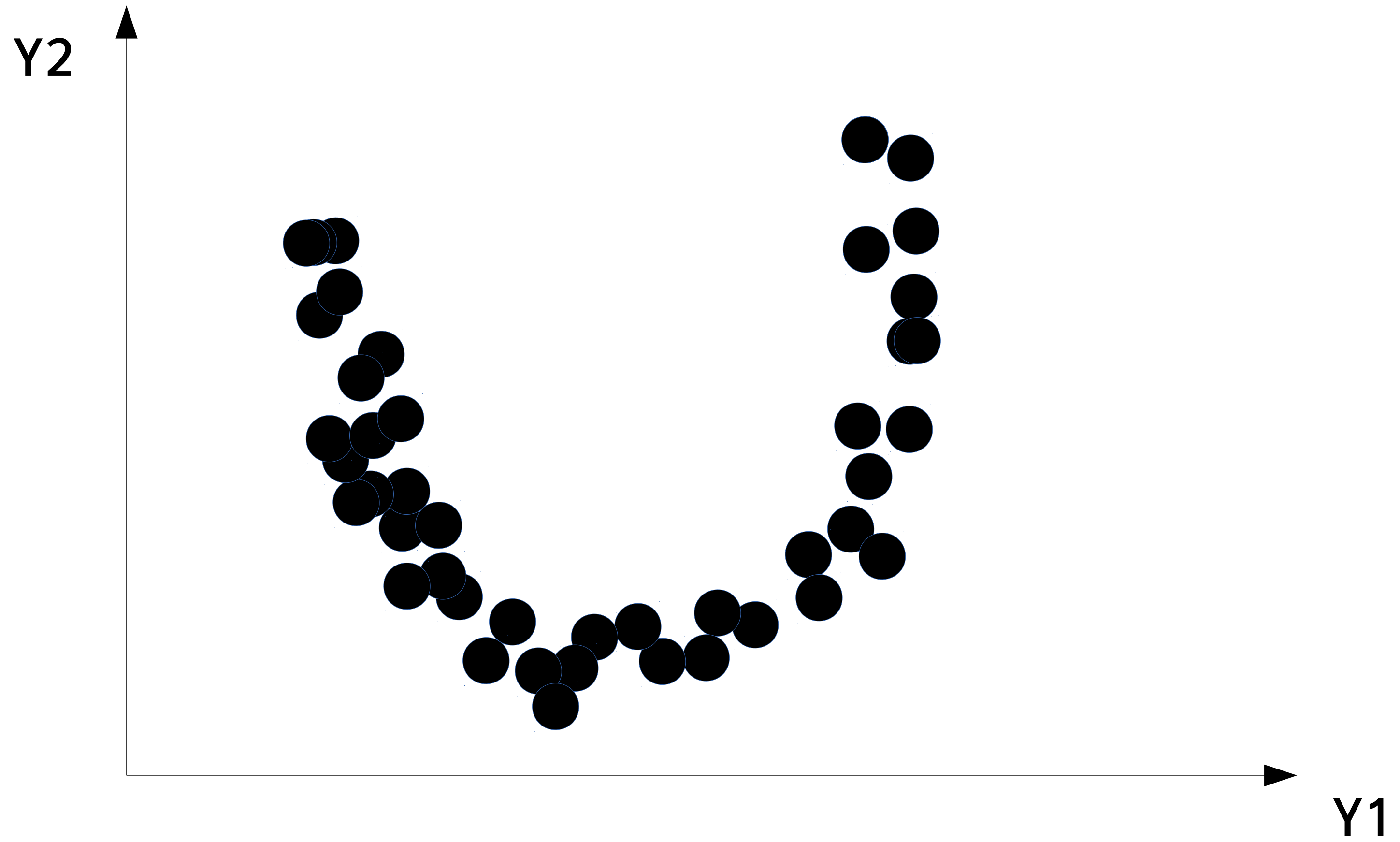


Unsupervised Learning

Energy-Based Unsupervised Learning

Y LeCun

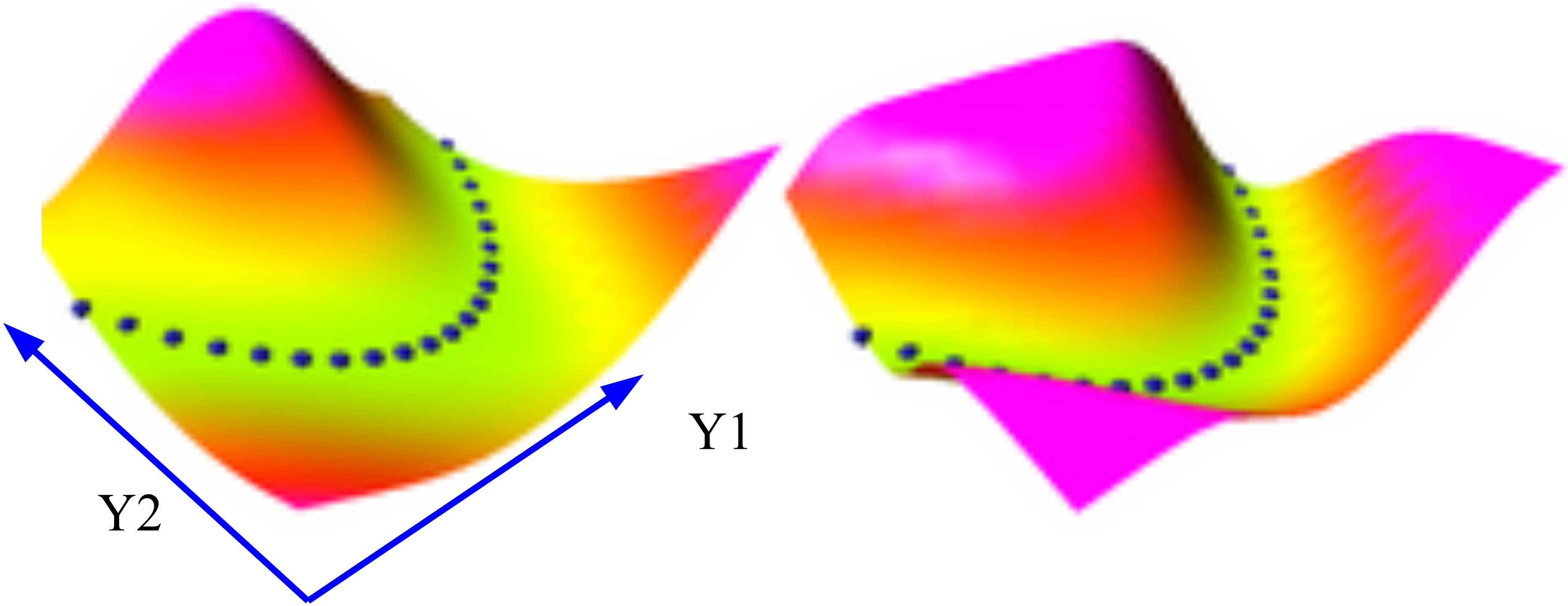
- **Learning an energy function (or contrast function) that takes**
  - Low values on the data manifold
  - Higher values everywhere else



Capturing Dependencies Between Variables with an Energy Function

- **The energy surface is a "contrast function" that takes low values on the data manifold, and higher values everywhere else**
  - Special case: energy = negative log density
  - Example: the samples live in the manifold

$$Y_2 = (Y_1)^2$$



Energy-Based Unsupervised Learning

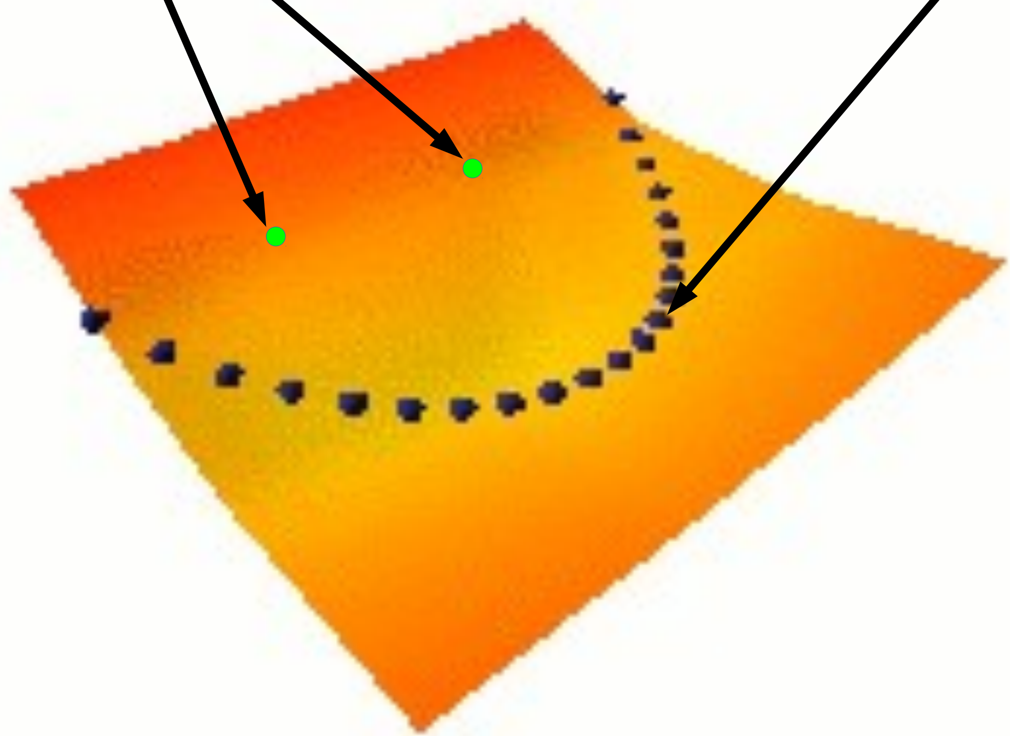
- **Energy Function:** Takes low value on data manifold, higher values everywhere else
- Push down on the energy of desired outputs. Push up on everything else.
- **But how do we choose where to push up?**

Implausible futures

(high energy)

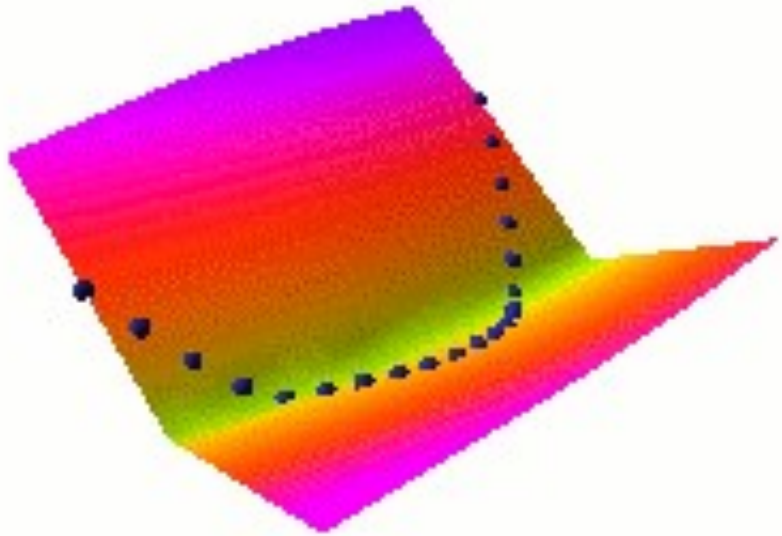
Plausible futures

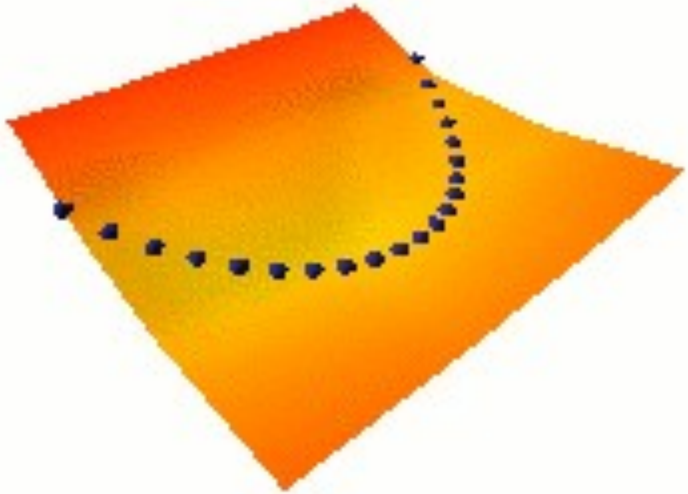
(low energy)



Learning the Energy Function

- **parameterized energy function E(Y,W)**
  - Make the energy low on the samples
  - Make the energy higher everywhere else
  - Making the energy low on the samples is easy
  - But how do we make it higher everywhere else?





Seven Strategies to Shape the Energy Function

- 1. build the machine so that the volume of low energy stuff is constant
  - PCA, K-means, GMM, square ICA
- 2. push down of the energy of data points, push up everywhere else
  - Max likelihood (needs tractable partition function)
- 3. push down of the energy of data points, push up on chosen locations
  - contrastive divergence, Ratio Matching, Noise Contrastive Estimation, Minimum Probability Flow
- 4. minimize the gradient and maximize the curvature around data points
  - score matching
- 5. train a dynamical system so that the dynamics goes to the manifold
  - denoising auto-encoder
- 6. use a regularizer that limits the volume of space that has low energy
  - Sparse coding, sparse auto-encoder, PSD
- 7. if E(Y) = ||Y - G(Y)||^2, make G(Y) as "constant" as possible.
  - Contracting auto-encoder, saturating auto-encoder

#1: constant volume of low energy
Energy surface for PCA and K-means

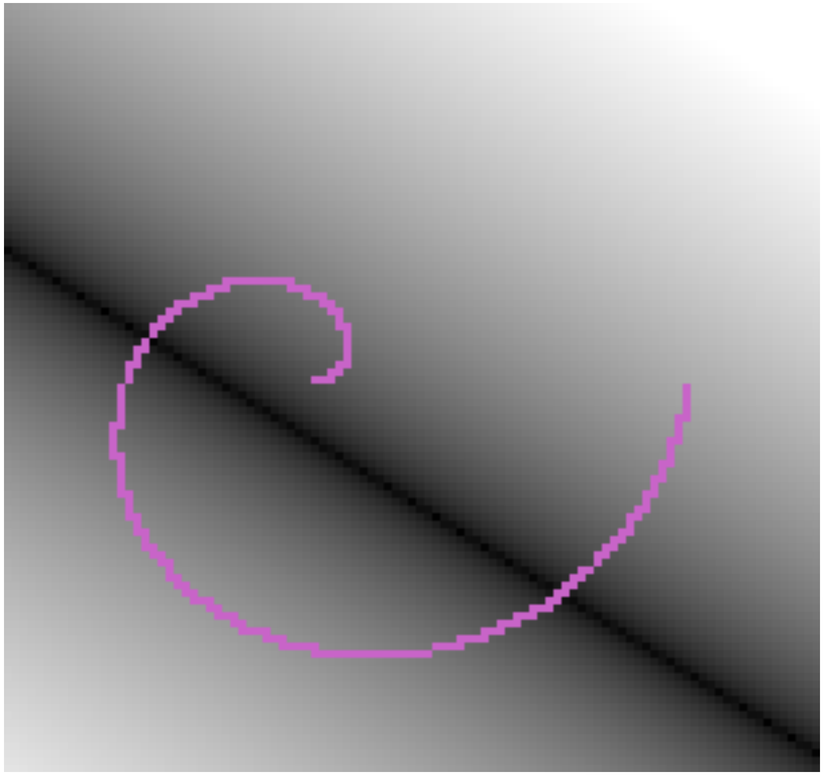
- **1. build the machine so that the volume of low energy stuff is constant**
  - PCA, K-means, GMM, square ICA...

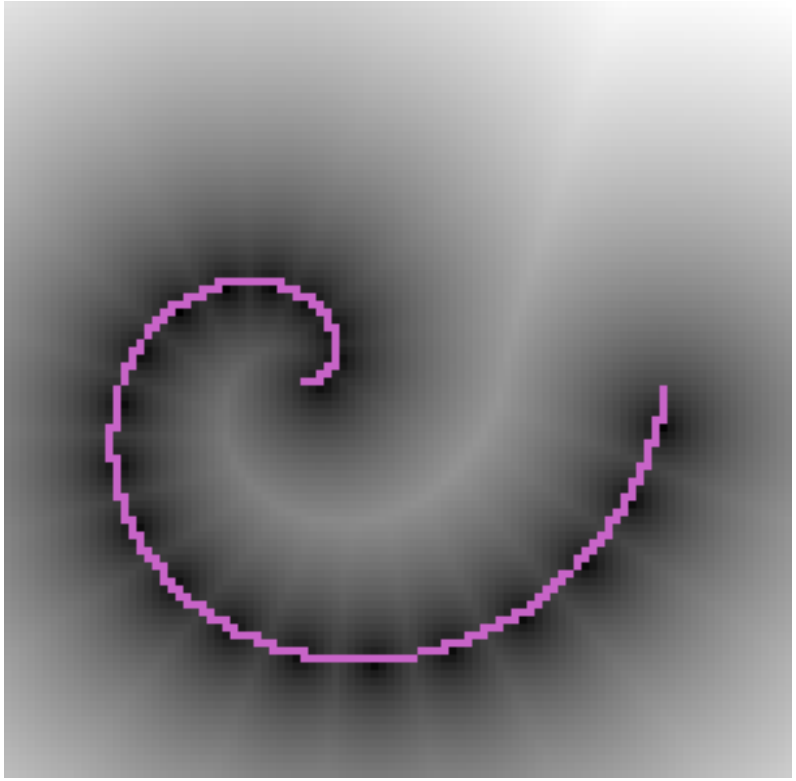
PCA

$$E(Y) = \|W^T W Y - Y\|^2$$

K-Means,
Z constrained to 1-of-K code

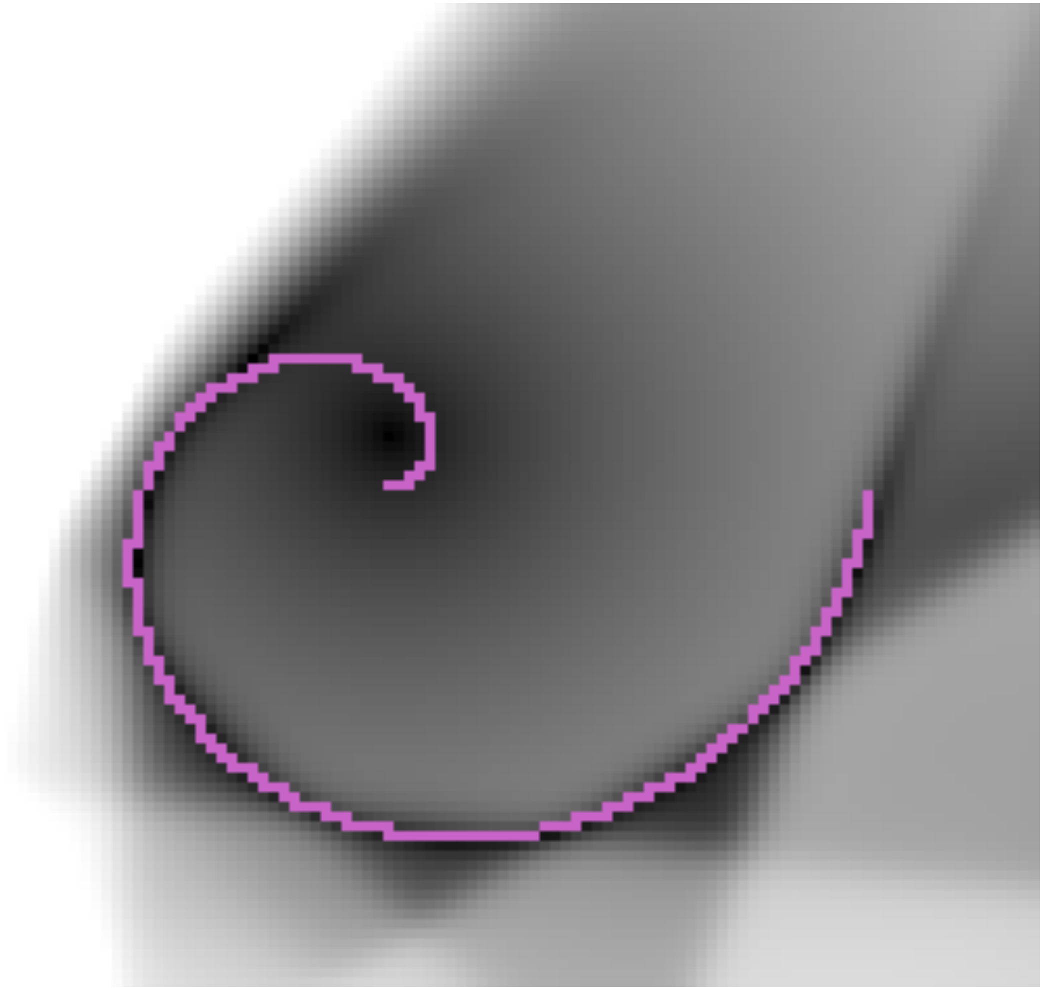
$$E(Y) = \min_z \sum_i \|Y - W_i Z_i\|^2$$





#6. use a regularizer that limits the volume of space that has low energy

- Sparse coding, sparse auto-encoder, Predictive Sparse Decomposition



”Why generative models” take-aways:

- Any energy-based unsupervised learning method can be seen as a probabilistic model by estimating the partition function
- I claim that any unsupervised learning method can be seen as energy-based, and can thus be transformed into a generative or probabilistic model
- Explicit probabilistic models are useful, because once we have one, we can use it “out of the box” for any of a variety of “common sense” tasks. No extra training or special procedures required.
  anomaly detection, denoising, filling in the blanks/super-resolution, compression / representation (inferring latent variables), scoring “realism” of samples, generating samples, ….

2. What is a variational autoencoder?

- Tutorial by Jaan Altosaar: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

“What is a VAE” take-aways:

DL interpretation:

- A VAE can be seen as a denoising compressive autoencoder
- Denoising = we inject noise to one of the layers. Compressive = the middle layers have lower capacity than the outer layers.

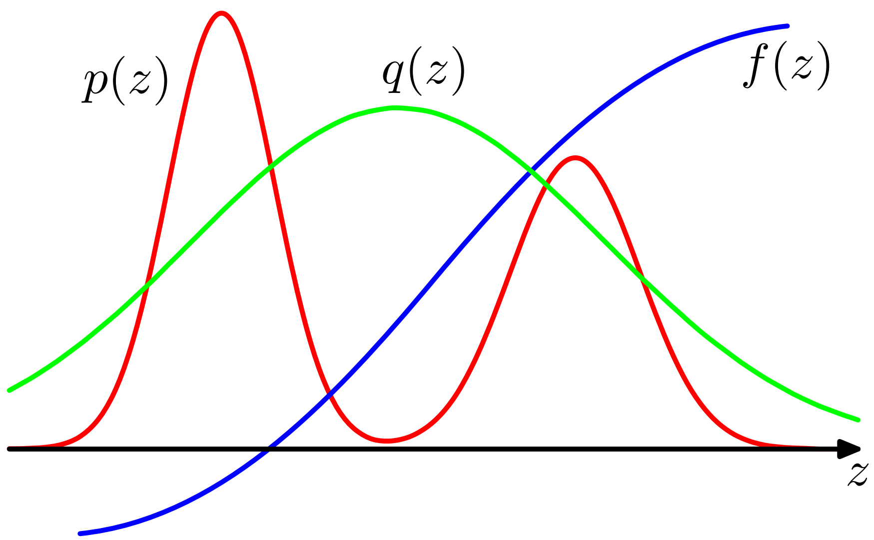
Probabilistic interpretation:

- The “decoder” of the VAE can be seen as a deep (high representational power) probabilistic model that can give us explicit likelihoods
- The “encoder” of the VAE can be seen as a variational distribution used to help train the decoder

2. From importance sampling to VAEs

- Selected slides from Shakir Mohamed's talk at the Deep Learning Summer School 2016

Importance Sampling



Integral problem $$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Proposal $$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}$$

Importance Weight $$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

Notation

Always think of *q(z|x)* but often will write *q(z)* for simplicity.

$$w^{(s)} = \frac{p(z)}{q(z)} \qquad z^{(s)} \sim q(z)$$

Conditions

- *q(z|x)*>0, when *f(z)p(z)* ≠ 0.
- Easy to sample from *q(z)*.

Monte Carlo $$p(\mathbf{x}) = \frac{1}{S}\sum_s w^{(s)} p(\mathbf{x}|\mathbf{z}^{(s)})$$

Importance Sampling to Variational Inference

Integral problem

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Proposal

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}$$

Importance Weight

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

Jensen's inequality

$$\log \int p(x)g(x)dx \geq \int p(x)\log g(x)dx$$

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z}) \log \left(p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})} \right) d\mathbf{z}$$

$$= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) - \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})}$$

Variational lower bound

$$\boxed{\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]}$$

Variational Free Energy

$$\mathcal{F}(\mathbf{x}, q) = \underbrace{\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction}} - \underbrace{KL[q(\mathbf{z})\|p(\mathbf{z})]}_{\text{Penalty}}$$

Approx. Posterior

Interpreting the bound:

- **Approximate posterior distribution** ***q(z|x)*****:** Best match to true posterior *p(z|x)*, one of the unknown inferential quantities of interest to us.

- **Reconstruction cost**: The expected log-likelihood measures how well samples from *q(z|x)* are able to explain the data *x*.

- **Penalty:** Ensures that the explanation of the data *q(z|x)* doesn't deviate too far from your beliefs *p(z)*. A mechanism for realising Ockham's razor.

Other Families of Variational Bounds

Variational Free Energy

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]$$

Multi-sample Variational Objective

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(z)}\left[\log \frac{1}{S}\sum_s \frac{p(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}|\mathbf{z})\right]$$

Renyi Variational Objective

$$\mathcal{F}(\mathbf{x}, q) = \frac{1}{1-\alpha}\mathbb{E}_{q(z)}\left[\left(\log \frac{1}{S}\sum_s \frac{p(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}|\mathbf{z})\right)^{1-\alpha}\right]$$

Other generalised families exist. Optimal solution is the same for all objectives.

“From importance sampling to VAE” take-aways:

- The VAE objective function can be derived in a way that I think is pretty unobjectionable to Bayesians and frequentists alike.
- Treat the decoder as a likelihood model we wish to train with maximum likelihood. We want to use importance sampling as $p(x|z)$ is low for most z.
- The encoder is a trainable importance sampling distribution, and the VAE objective is a lower bound to the likelihood by Jensen’s inequality.