Clasificadores Probabilísticos en Aprendizaje Automático

Modelos Probabilísticos Profundos

Daniel Ramos Castro

Contribuciones de Juan Maroñas Molano (Doctorando Univ. Politécnica Valencia)

daniel.ramos@uam.es

Audias – Audio, Data Intelligence and Speech

Universidad Autónoma de Madrid

http://audias.ii.uam.es

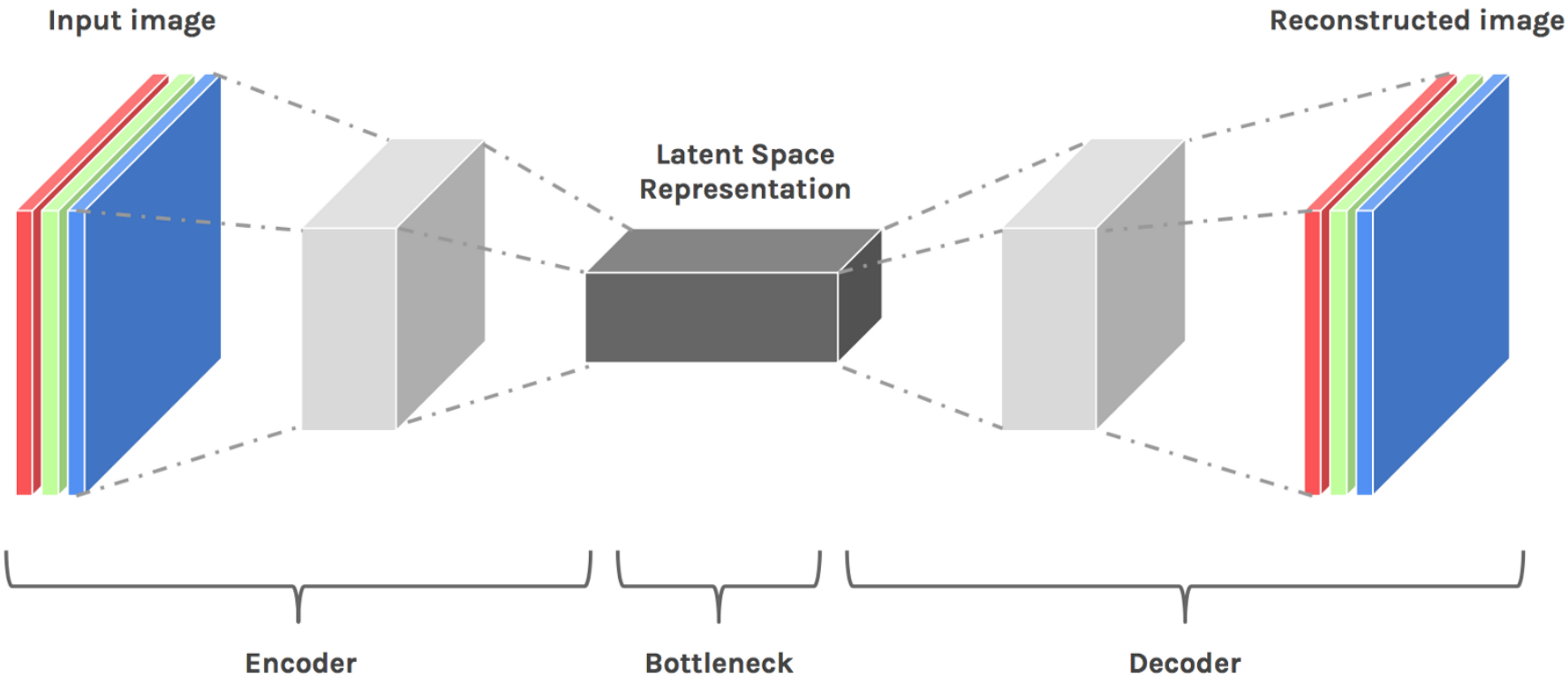
Sumario del Día

- Modelos probabilísticos profundos
  - Autoencoder Variacional
  - Redes Neuronales Profundas

- Teoría de la Decisión

Autoencoder Variacional
(*Variational Autoencoder, VAE*)

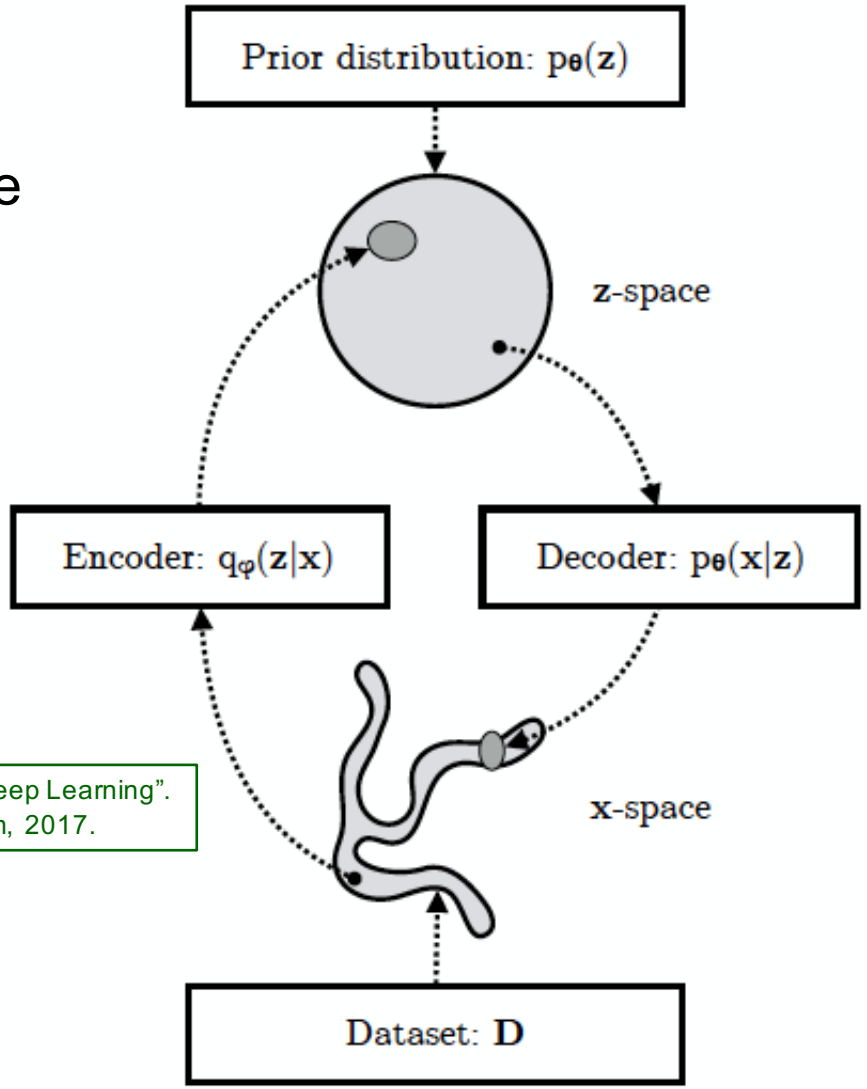
Autoencoder Variacional: Interpretación Neuronal



https://datasciencevision.com/autoencoders/

Autoencoder Variacional (VAE)

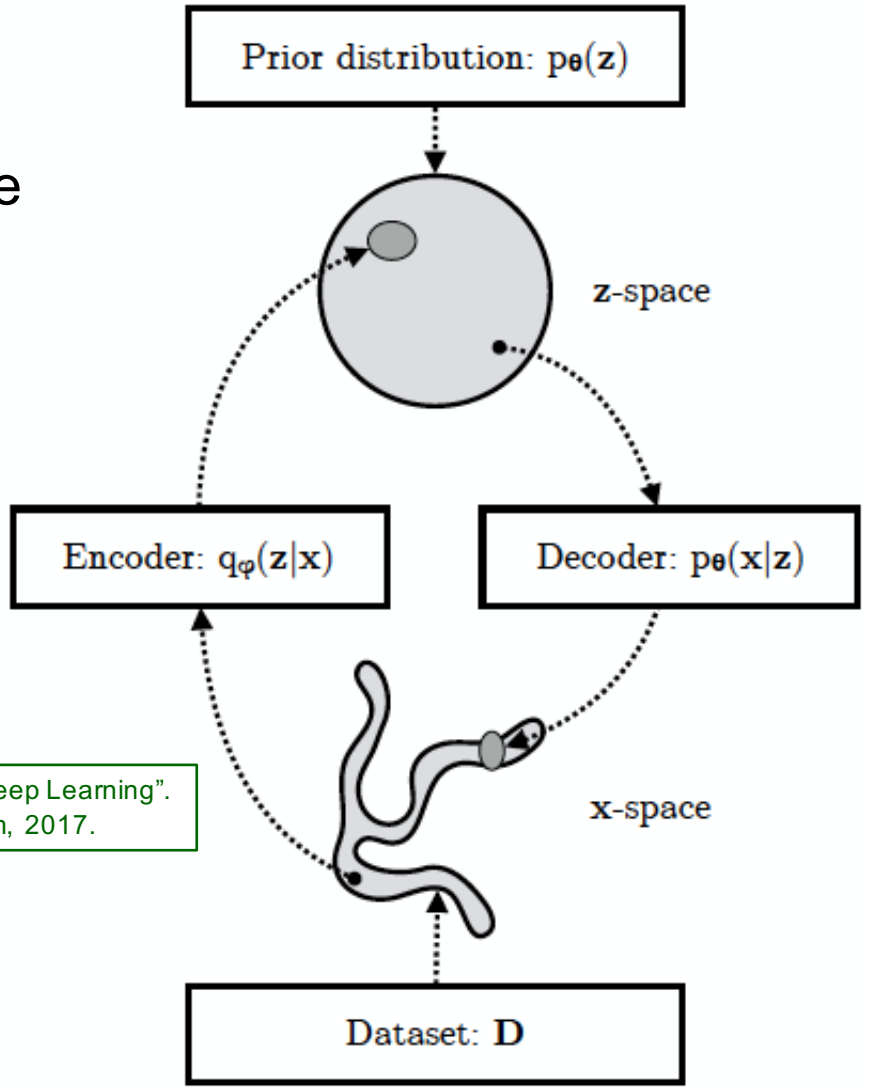
- Modelo probabilístico generativo
- Interpretación neuronal: 2 pasos:
  - Transformar datos a un espacio latente
    - Encoder (NN con pesos φ)
    - Devuelve parámetros
  - Transformar del espacio latente de nuevo al espacio observado de forma probabilística
    - *Decoder* (NN con pesos θ)
    - Devuelve parámetros
- Objetivo
  - Reducir al mínimo el error de reconstrucción
    - Para que la salida del VAE se parezca al máximo a su entrada



D. Kingma. "Variational Inference and Deep Learning". PhD Thesis, Univ. Van Amsterdam, 2017.

Autoencoder Variacional (VAE)

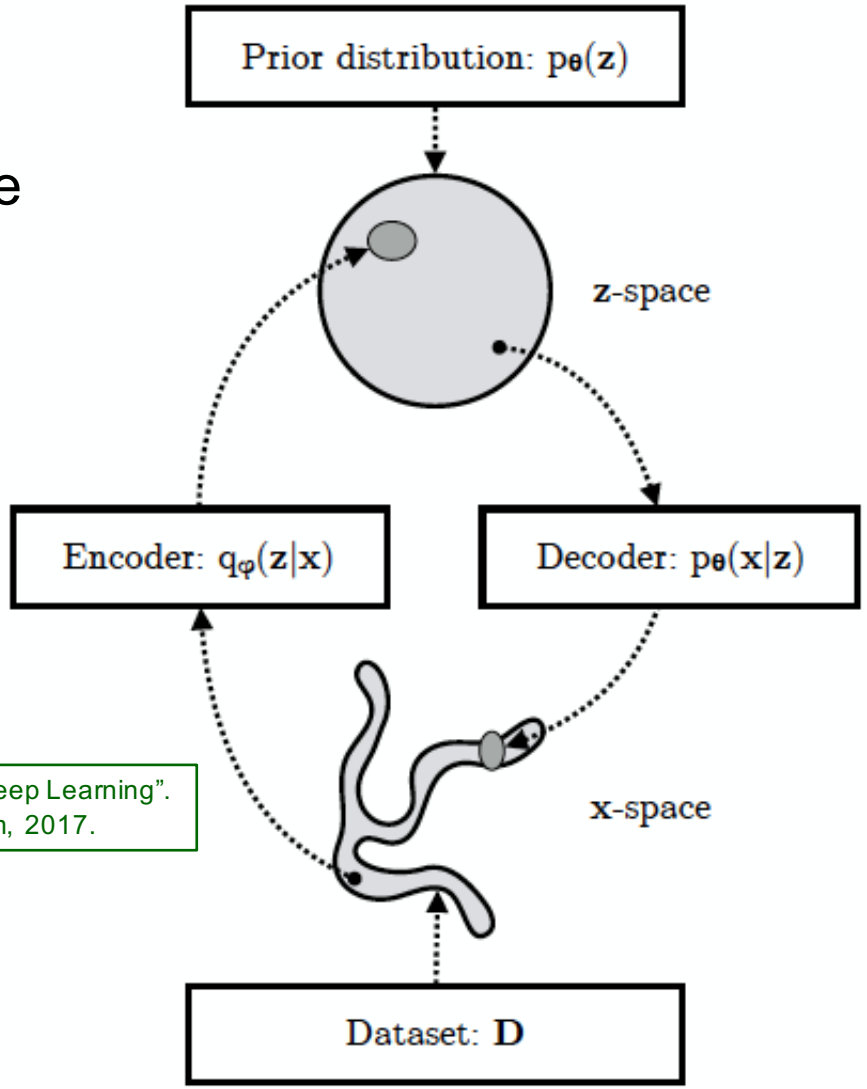
- Modelo probabilístico generativo
- Interpretación neuronal: 2 pasos:
  - Transformar datos a un espacio latente
    - Encoder (NN con pesos φ)
    - Devuelve parámetros



D. Kingma. "Variational Inference and Deep Learning". PhD Thesis, Univ. Van Amsterdam, 2017.

Autoencoder Variacional (VAE)

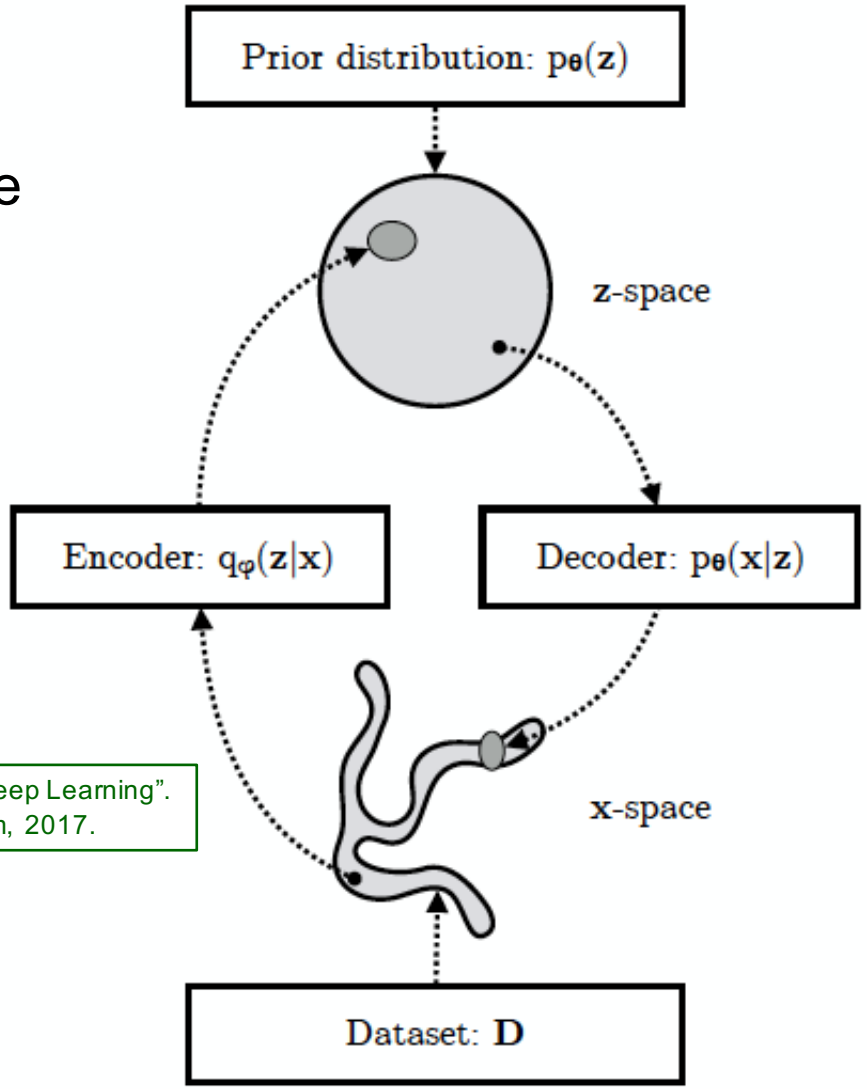
- Modelo probabilístico generativo
- Interpretación neuronal: 2 pasos:
  - Transformar datos a un espacio latente
    - Encoder (NN con pesos φ)
    - Devuelve parámetros
  - Transformar del espacio latente de nuevo al espacio observado de forma probabilística
    - *Decoder* (NN con pesos θ)
    - Devuelve parámetros



D. Kingma. "Variational Inference and Deep Learning". PhD Thesis, Univ. Van Amsterdam, 2017.

Autoencoder Variacional (VAE)

- Modelo probabilístico generativo
- Interpretación neuronal: 2 pasos:
  - Transformar datos a un espacio latente
    - Encoder (NN con pesos φ)
    - Devuelve parámetros
  - Transformar del espacio latente de nuevo al espacio observado de forma probabilística
    - *Decoder* (NN con pesos θ)
    - Devuelve parámetros
- Objetivo
  - Reducir al mínimo el error de reconstrucción
    - Para que la salida del VAE se parezca al máximo a su entrada



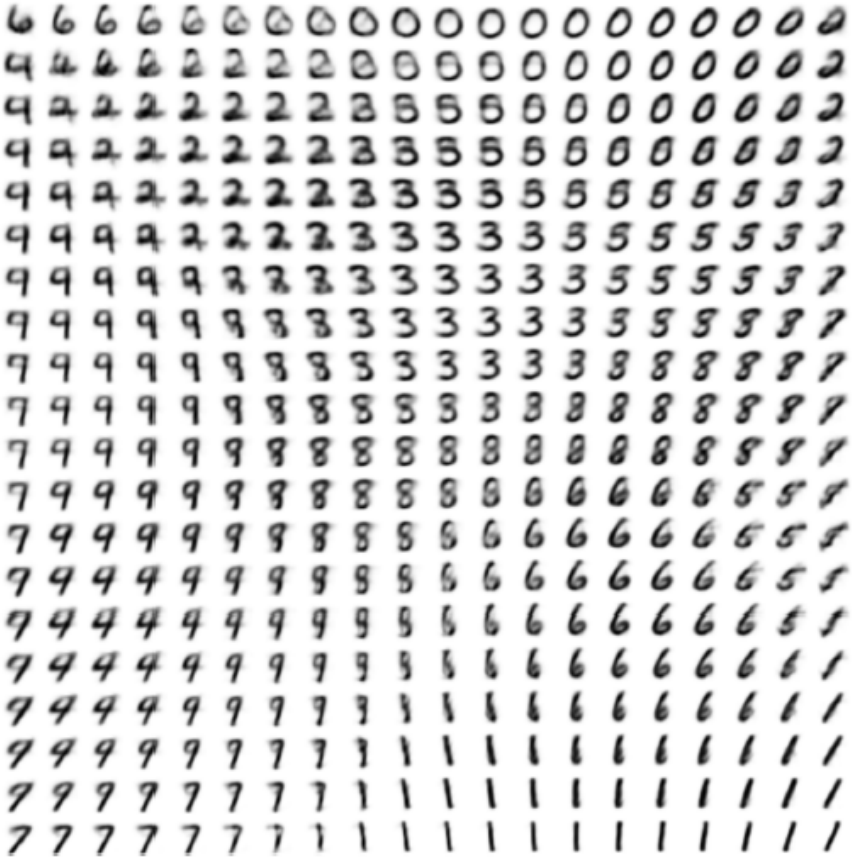
D. Kingma. "Variational Inference and Deep Learning". PhD Thesis, Univ. Van Amsterdam, 2017.

Autoencoder Variacional

- ¿Para qué?
  - Representación probabilística simplificada de los datos
  - A partir de dicha representación, generar nuevos datos nunca vistos antes

D. Kingma, M. Welling. "Auto-Encoding Variational Bayes". Arxiv, 2014.

(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Autoencoder Variacional

- Variables observadas: $\boldsymbol{x}$
  - Por ejemplo, vector de píxeles de dígitos MNIST

Autoencoder Variacional

- Variables observadas: $\boldsymbol{x}$
  - Por ejemplo, vector de píxeles de dígitos MNIST
- Variables latentes: $\boldsymbol{z}$
  - Un espacio de dimensión mucho menor

Autoencoder Variacional

- Variables observadas: $\boldsymbol{x}$
  - Por ejemplo, vector de píxeles de dígitos MNIST
- Variables latentes: $\boldsymbol{z}$
  - Un espacio de dimensión mucho menor
- Objetivo: modelo generativo (θ: pesos de una NN)
  - $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z})$

Autoencoder Variacional

- Variables observadas: $\boldsymbol{x}$
  - Por ejemplo, vector de píxeles de dígitos MNIST
- Variables latentes: $\boldsymbol{z}$
  - Un espacio de dimensión mucho menor
- Objetivo: modelo generativo (θ: pesos de una NN)
  - $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z})$
- Generación de datos
  - Primero muestreo un dato latente
    - $\boldsymbol{z}^{(i)} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$
  - Luego obtengo el dato observado
    - $\boldsymbol{x}^{(i)} \sim p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}^{(i)}\right)$

Autoencoder Variacional

D. Kingma, M. Welling. “Auto-Encoding Variational Bayes”. Arxiv, 2014.

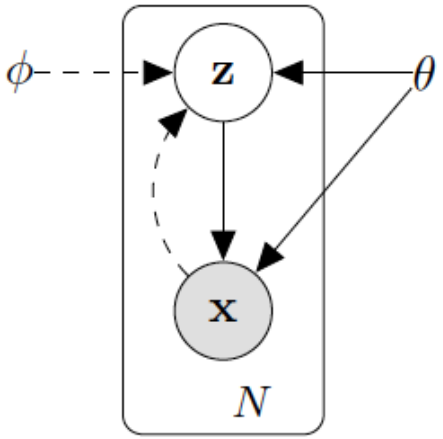


Figure 1: The type of directed graphical model under consideration. Solid lines denote the generative model $p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, dashed lines denote the variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ to the intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. The variational parameters $\boldsymbol{\phi}$ are learned jointly with the generative model parameters $\boldsymbol{\theta}$.

Autoencoder Variacional

- Probabilidad a priori $p_{\boldsymbol{\theta}}(\boldsymbol{z})$
  - Suele ser una gaussiana estándar multivariada (no depende de $\boldsymbol{\theta}$)
  - Representación simplificada de los datos, probabilística
  - Ejemplo: Distintos dígitos MNIST deberían ubicarse en distintas zonas del espacio latente

Autoencoder Variacional

- Probabilidad a priori $p_{\boldsymbol{\theta}}(\boldsymbol{z})$
  - Suele ser una gaussiana estándar multivariada (no depende de $\boldsymbol{\theta}$)
  - Representación simplificada de los datos, probabilística
  - Ejemplo: Distintos dígitos MNIST deberían ubicarse en distintas zonas del espacio latente
- *Likelihood* $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$
  - Ejemplo dígitos NIST: Bernoulli multivariada
    - $p_{\boldsymbol{\theta}}\left(\boldsymbol{x}|\boldsymbol{z}^{(i)}\right)$ devuelve la probabilidad de que cada píxel sea 0 ó 1 (valor del parámetro Bernoulli)
      - Dependiendo del valor de la variable latente $\boldsymbol{z}^{(i)}$
    - NN cuya entrada es $\boldsymbol{z}^{(i)}$ y devuelve parámetros de Bernoulli
      - Sus pesos son $\boldsymbol{\theta}$

Autoencoder Variacional: Inferencia

- Necesitamos conocer el *encoder*

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{\int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})dZ}$$ ¡Intratable!

Autoencoder Variacional: Inferencia

- Necesitamos conocer el *encoder*

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{\int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})dZ}$$ ¡Intratable!

- Solución: inferencia variacional
  - Tomo una distribución $q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{x})$ para aproximar $p_{\theta}(\boldsymbol{z}|\boldsymbol{x})$
    - Gaussiana, componentes independientes:
      - $q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) = \mathcal{N}\left(\boldsymbol{z}|\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\right)$
  - $\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}$ se obtienen de una NN con entrada $\boldsymbol{x}^{(i)}$
    - Con pesos $\boldsymbol{\varphi}$

Autoencoder Variacional: Inferencia

- Minimizar una distancia entre $p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ y $q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$
  - Optimizo $\boldsymbol{\varphi}$ y $\boldsymbol{\theta}$, y obtengo el mínimo de esta distancia
    - Típicamente, distancia Kullback-Leibler (KL)

$$D_{KL}\left(q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)\right)$$ ¡Intratable!

Autoencoder Variacional: Inferencia

- Minimizar una distancia entre $p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ y $q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$
  - Optimizo $\boldsymbol{\varphi}$ y $\boldsymbol{\theta}$, y obtengo el mínimo de esta distancia
    - Típicamente, distancia Kullback-Leibler (KL)

$$D_{KL}\left(q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)\right)$$ ¡Intratable!

- Operando…

¡Intratable!

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

¡Tratable!

Evidence Lower Bound (ELBO)

Autoencoder Variacional: Inferencia

- Minimizar una distancia entre $p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ y $q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$
  - Optimizo $\boldsymbol{\varphi}$ y $\boldsymbol{\theta}$, y obtengo el mínimo de esta distancia
    - Típicamente, distancia Kullback-Leibler (KL)

$$D_{KL}\left(q_{\boldsymbol{\varphi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)\right)$$ ¡Intratable!

- Operando…

¡Intratable!

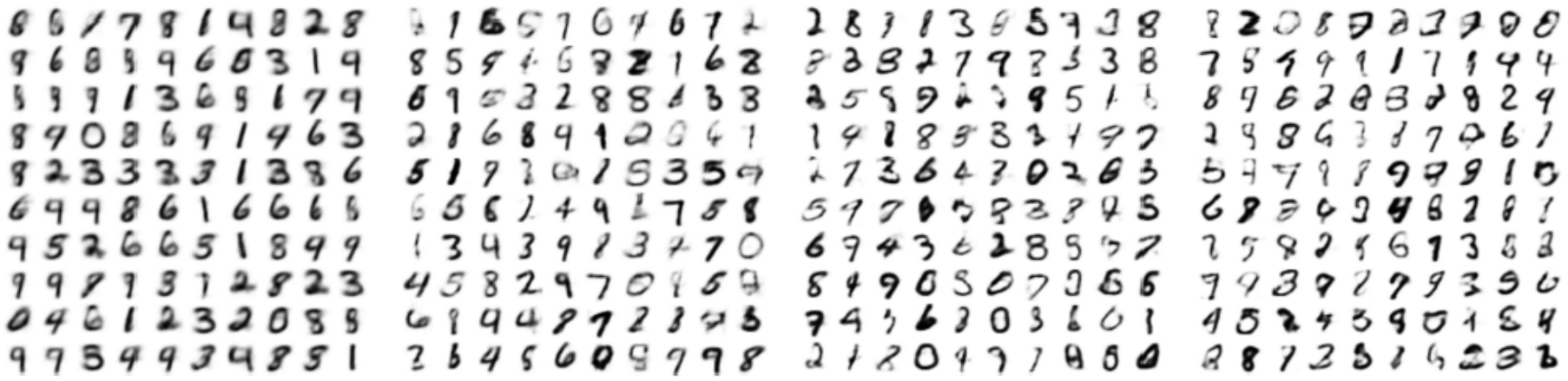
$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

¡Tratable!

Evidence Lower Bound (ELBO)

  - Problema alternativo: maximizar ELBO
    - Aumenta verosimilitud marginal, minimiza la distancia KL ¡simultáneamente!
      - Algoritmo Auto Encoding Variational Bayes
      - Reparameterization trick

Autoencoder Variacional: Ejemplo

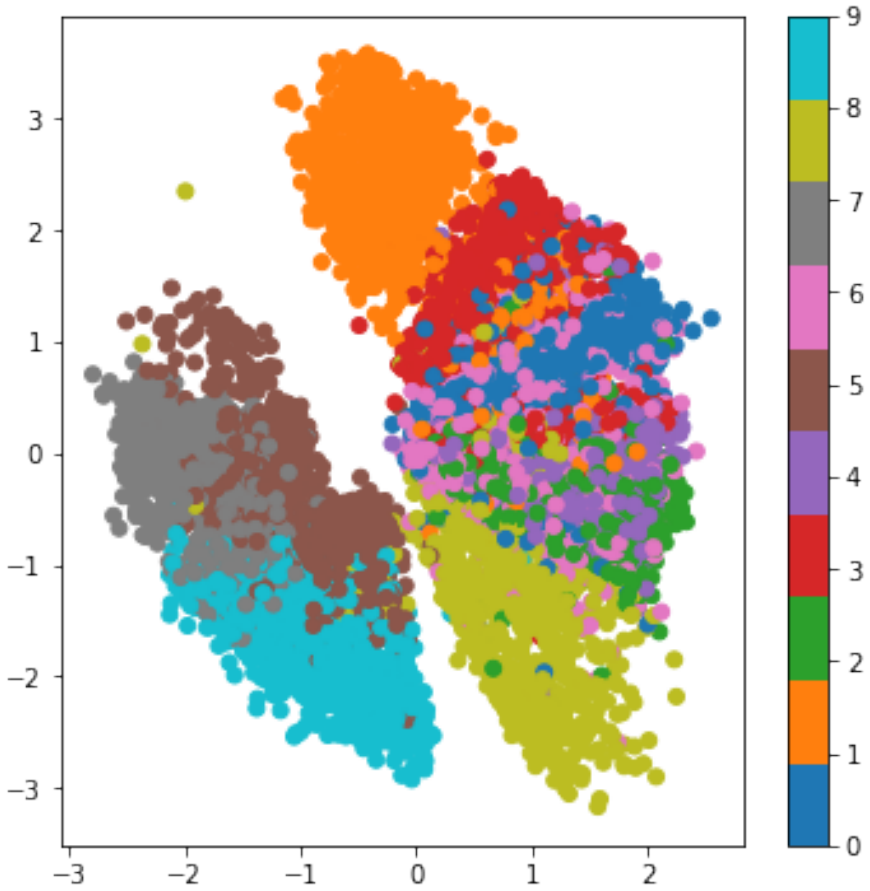


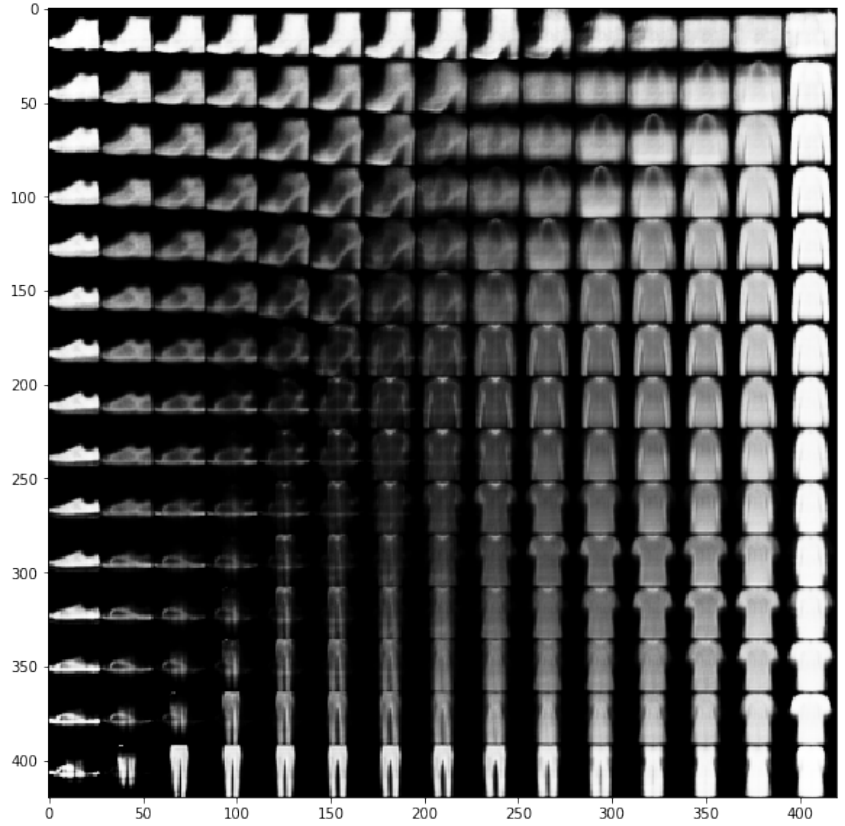
(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

D. Kingma, M. Welling. “Auto-Encoding Variational Bayes”. Arxiv, 2014.

Autoencoder Variacional: Ejemplo





https://datasciencevision.com/autoencoders/

Redes Neuronales Bayesianas (BNN)

Redes Neuronales

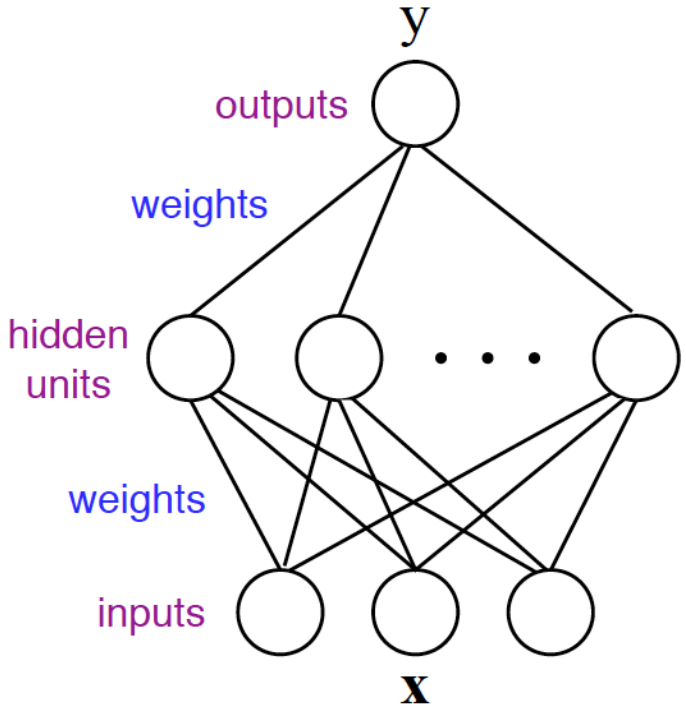
- Datos:

$$\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N} = (X, \mathbf{y})$$

- Pesos: $\boldsymbol{\theta}$
- *Likelihood*:

$$p(y^{(n)}|\mathbf{x}^{(n)}, \boldsymbol{\theta})$$

  - Modelada como perceptrón multicapa



Z. Ghahramani. “A history of Bayesian Neural Networks”. NIPS 2016 Keynote Speech Slides.

Redes Neuronales: Optimización

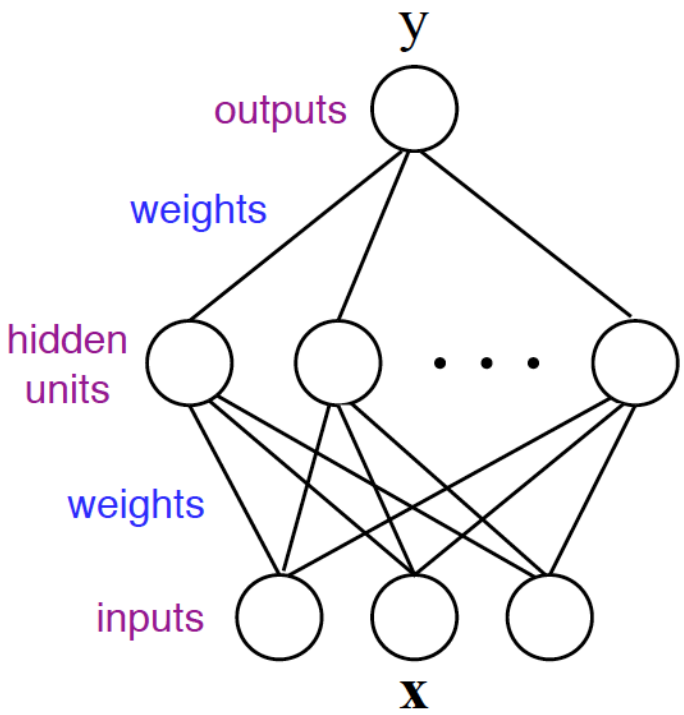
- *Backpropagation,* Stochastic Gradient Descent (SGD)
  - Se entrena la red con las salidas conocidas que debería obtener (ground-truth)
  - Se utiliza una función objetivo entre lo que la red saca y lo que debería sacar (*cross-entropy*, entropía cruzada)
  - Se calcula el gradiente para optimizar la función de coste
  - Se propagan dichos gradientes hacia atrás capa a capa
  - Finalmente, el óptimo es el mínimo de dicha función de coste
- Problema de ajuste puntual
  - *Point-estimate*
  - ML o MAP, típicamente

Redes Neuronales Bayesianas: Inferencia

- Las BNNs pretenden generalizar las redes neuronales
  - Los pesos en realidad no se optimizan hasta obtener el máximo
  - En realidad, los pesos se consideran no conocidos
  - Se intenta obtener un modelo probabilístico de los pesos, las entradas y las salidas

$$\begin{aligned}
&\text{prior} && p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \\
&\text{posterior} && p(\boldsymbol{\theta}|\boldsymbol{\alpha}, \mathcal{D}) \propto p(\mathbf{y}|X, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \\
&\text{prediction} && p(y'|\mathcal{D}, \mathbf{x}', \boldsymbol{\alpha}) = \int p(y'|\mathbf{x}', \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha})\, d\boldsymbol{\theta}
\end{aligned}$$

Z. Ghahramani. "A history of Bayesian Neural Networks". NIPS 2016 Keynote Speech Slides.



Redes Neuronales Bayesianas: Inferencia

- El posterior es muy difícil de obtener
  - Problema claramente no tratable
- Inferencia variacional: aproximación muy simplista

Redes Neuronales Bayesianas: Inferencia

- El posterior es muy difícil de obtener
  - Problema claramente no tratable
- Inferencia variacional: aproximación muy simplista
- Solución: métodos Montecarlo
  - Muestreamos el posterior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}|\boldsymbol{\alpha}, \mathcal{D})$$

  - Aproximamos la integral

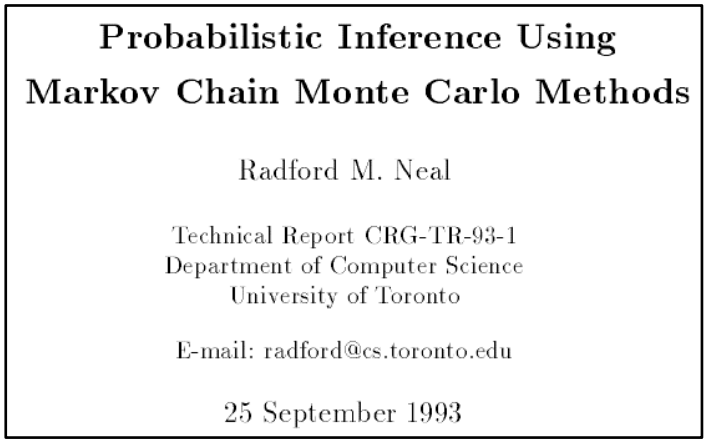
$$\int p(y'|x', \boldsymbol{\theta})\, p(\boldsymbol{\theta}|\boldsymbol{\alpha}, \mathcal{D}) d\boldsymbol{\theta} \approx \frac{1}{M} \sum_{i=1}^{M} p\left(y'|x', \boldsymbol{\theta}^{(i)}\right)$$

Redes Neuronales Bayesianas: Inferencia

- Métodos Markov Chain Monte Carlo
  - Paseo aleatorio para recorrer el posterior
    - Deben hacerlo rápido
    - Deben asegurar que se recorre bien
      - Las muestras se distribuyen como el posterior

Redes Neuronales Bayesianas: Inferencia

- Métodos Markov Chain Monte Carlo
  - Paseo aleatorio para recorrer el posterior
    - Deben hacerlo rápido
    - Deben asegurar que se recorre bien
      - Las muestras se distribuyen como el posterior
  - Algoritmos más populares
    - Metropolis-Hastings
      - Muy util con distribuciones simples
      - No es el caso de las BNN
    - Hamiltonian Monte Carlo (HMC)
      - Mucho más adecuado para BNNs



Probabilistic Inference Using Markov Chain Monte Carlo Methods

Radford M. Neal

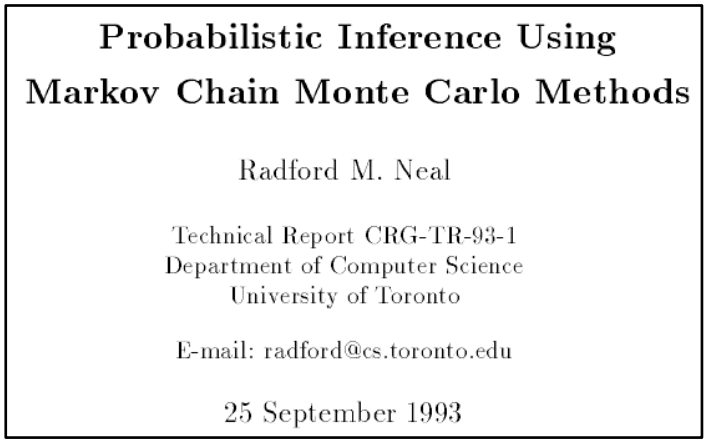
Technical Report CRG-TR-93-1
Department of Computer Science
University of Toronto

E-mail: radford@cs.toronto.edu

25 September 1993

Redes Neuronales Bayesianas: Inferencia

- Métodos Markov Chain Monte Carlo
  - Paseo aleatorio para recorrer el posterior
    - Deben hacerlo rápido
    - Deben asegurar que se recorre bien
      - Las muestras se distribuyen como el posterior
  - Algoritmos más populares
    - Metropolis-Hastings
      - Muy util con distribuciones simples
      - No es el caso de las BNN
    - Hamiltonian Monte Carlo (HMC)
      - Mucho más adecuado para BNNs
  - Ejemplo (David Duvenaud):
    - https://www.youtube.com/watch?v=Vv3f0QNWvWQ



Probabilistic Inference Using Markov Chain Monte Carlo Methods

Radford M. Neal

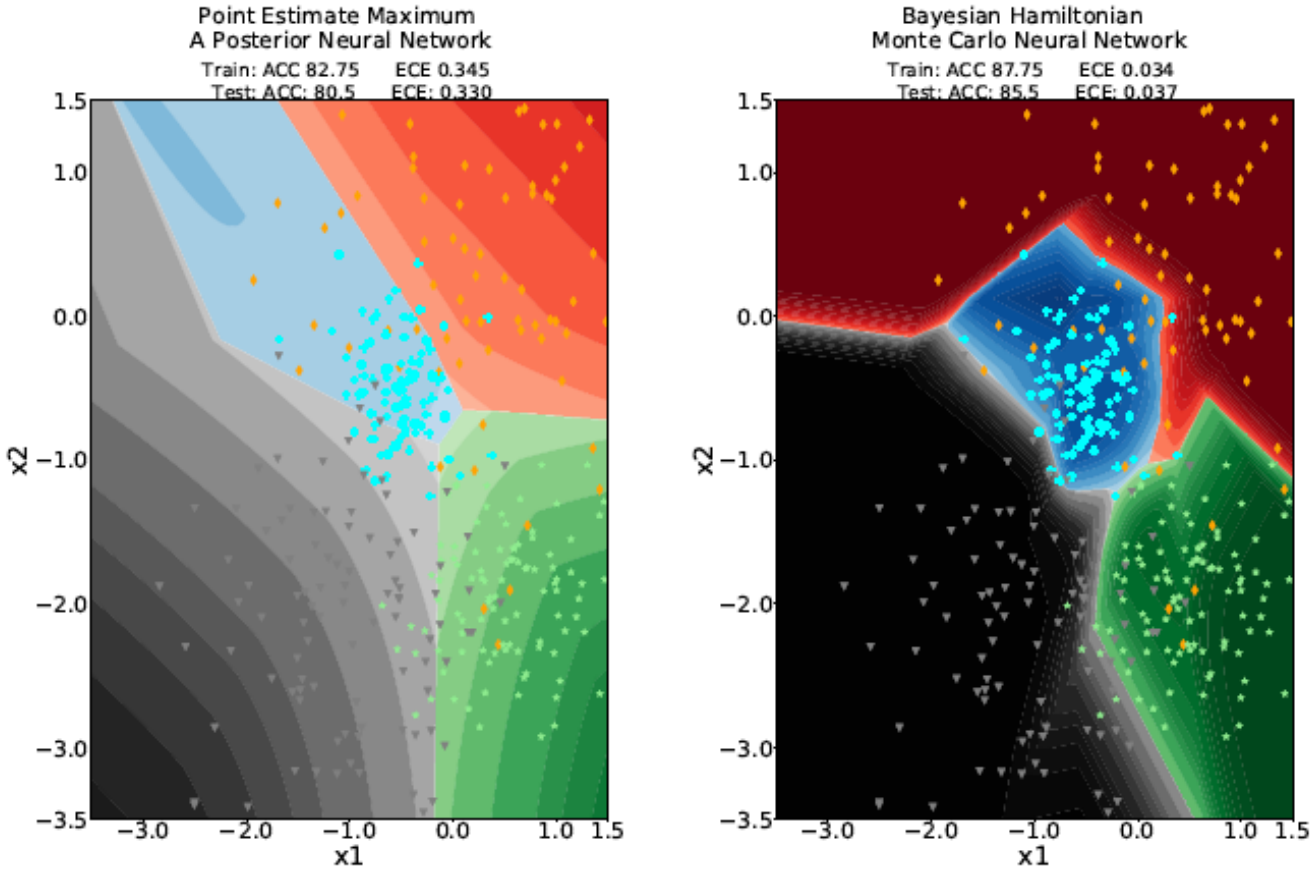
Technical Report CRG-TR-93-1
Department of Computer Science
University of Toronto

E-mail: radford@cs.toronto.edu

25 September 1993

Redes Neuronales Bayesianas

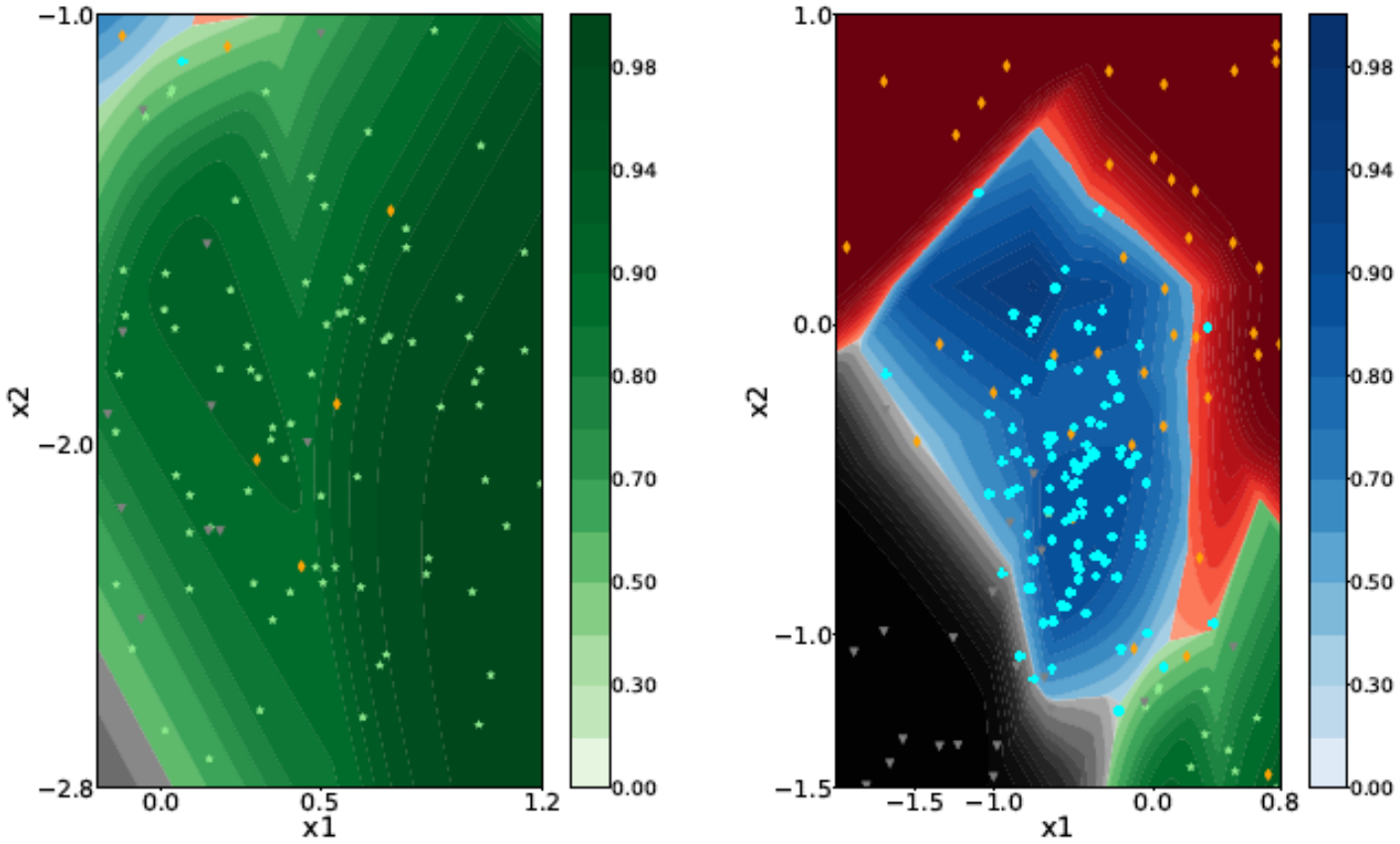
- En tareas simples, BNNs con HMC pueden superar a las NNs MAP



- Problema: las tareas actuales son muy complejas
  - Cuesta muchísimo que las BNN superen a las DNN

Redes Neuronales Bayesianas

- En tareas simples, BNNs con HMC pueden superar a las NNs MAP



- Problema: las tareas actuales son muy complejas
  - Cuesta muchísimo que las BNN superen a las DNN

Clasificadores Probabilísticos en Aprendizaje Automático

Modelos Probabilísticos Profundos

Daniel Ramos Castro

Contribuciones de Juan Maroñas Molano (Doctorando Univ. Politécnica Valencia)

daniel.ramos@uam.es

Audias – Audio, Data Intelligence and Speech

Universidad Autónoma de Madrid

http://audias.ii.uam.es