Bayesian Deep Learning

Mauricio A. Álvarez PhD,
H.F. Garcia C. Guarnizo (TA)

Universidad Tecnológica de Pereira, Pereira, Colombia

Outline

1. **Bayesian Neural Networks**

Outline

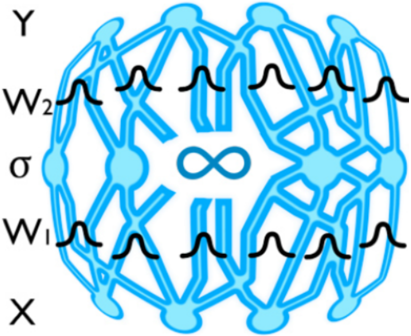
1. **Bayesian Neural Networks**

Motivation I

- Ubicar priors en los pesos de la red neurnal $p(\mathbf{W}_i)$:

$$\mathbf{W}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{1}$$

para $i \leq L$ (tal que $\boldsymbol{\omega} := \{\mathbf{W}_i\}_{i=1}^{L}$).



Motivation II

- La salida es una r.v.
  $\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}) = \mathbf{W}_L \sigma \left(\ldots \mathbf{W}_2 \sigma \left(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1 \right) \ldots \right)$
- Softmax likelihood para classificación:
  $p(y|\mathbf{x}, \omega) = \text{softmax}(\mathbf{f}(\mathbf{x}, \omega))$
- Gaussian para regresión: $p(\mathbf{y}|\mathbf{x}, \omega) = \mathcal{N}\left(\mathbf{y}; \mathbf{f}(\mathbf{x}, \omega), \tau^{-1}\mathbf{I}\right)$
- Pero calcular el posterior es difícil

$$p(\omega|\mathbf{X}, \mathbf{Y})$$

Deep learning como *approx. inference* I

Approximate inference in Bayesian NNs

- Definir $q_\theta(\omega)$ para aproximar el posterior $p(\omega|\mathbf{X},\mathbf{Y})$
- Se utiliza la divergencia KL para minimizar

$$
\begin{aligned}
&\mathrm{KL}\left(q_\theta(\omega)\|p(\omega|\mathbf{X},\mathbf{Y})\right)\\
&\propto \left[-\int q_\theta(\omega)\log p(\mathbf{Y}|\mathbf{X},\omega)\mathrm{d}\omega\right] + \mathrm{KL}\left(q_\theta(\omega)\|p(\omega)\right)\\
&=: \mathcal{L}(\theta)
\end{aligned}
$$

- Se aproxima la integral con *MC integration* $\widehat{\omega} \sim q_\theta(\omega)$:

$$\widehat{\mathcal{L}}(\theta) := -\log p(\mathbf{Y}|\mathbf{X},\widehat{\omega}) + \mathrm{KL}\left(q_\theta(\omega)\|p(\omega)\right)$$

Deep learning como *approx. inference* II

Inferencia aproximada estocástica en Bayesian NNs

- Unbiased estimator:

$$E_{\widehat{\omega} \sim q_\theta(\omega)}(\widehat{\mathcal{L}}(\theta)) = \mathcal{L}(\theta)$$

- Converge al mismo óptimo que $\mathcal{L}(\theta)$
- Ej. repetir:
  - muestrear de $\widehat{\omega} \sim q_\theta(\omega)$
  - And minimise (one step)

$$\widehat{\mathcal{L}}(\theta) = -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \mathrm{KL}\left(q_\theta(\omega) \| p(\omega)\right)$$

w.r.t θ

Deep learning como *approx. inference* III

Especificando $q_\theta(\cdot)$

- Dada la r.v. Bernoulli $\mathbf{z}_{i,j}$ y los parámetros variacionales $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (conjunto de matrices):

$$\mathbf{z}_{i,j} \sim \text{ Bernoulli } (p_i) \text{ for } i = 1, \ldots, L, j = 1, \ldots, K_{i-1}$$

$$\mathbf{W}_i = \mathbf{M}_i \cdot \operatorname{diag}\left([\mathbf{z}_{i,j}]_{j=1}^{K_i}\right)$$

$$q_\theta(\omega) = \prod q_{\mathbf{M}_i}(\mathbf{W}_i)$$

Summary

Minimizar la divergencia entre $q_\theta(\omega)$ and $p(\omega|\mathbf{X},\mathbf{Y})$:

- Repetir:
  - Muestrear $\widehat{\mathbf{z}}_{i,j} \sim$ Bernoulli (p_i) y hacer:

$$\widehat{\mathbf{W}}_i = \mathbf{M}_i \cdot \operatorname{diag}\left([\widehat{\mathbf{z}}_{i,j}]_{j=1}^{K_i}\right)$$

$$\widehat{\omega} = \left\{\widehat{\mathbf{W}}_i\right\}_{i=1}^{L}$$

  - = Hacer aleatoriamente columnas de $\mathbf{M}_i$ cero
  - Establecer aleatoriamente las neuronas de la red a cero
  - Minimizar

$$\widehat{\mathcal{L}}(\theta) = -\log p(\mathbf{Y}|\mathbf{X},\widehat{\omega}) + \mathrm{KL}\left(q_\theta(\omega)\|p(\omega)\right)$$

w.r.t. $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (conjunto de matrices)

Summary

Minimizar la divergencia entre $q_\theta(\omega)$ and $p(\omega|\mathbf{X}, \mathbf{Y})$:

- Repetir:
  - Muestrear $\widehat{\mathbf{z}}_{i,j} \sim \text{Bernoulli}\,(p_i)$ y hacer:

$$\widehat{\mathbf{W}}_i = \mathbf{M}_i \cdot \text{diag}\left([\widehat{\mathbf{z}}_{i,j}]_{j=1}^{K_i}\right)$$

$$\widehat{\omega} = \left\{\widehat{\mathbf{W}}_i\right\}_{i=1}^{L}$$

  - = Hacer aleatoriamente columnas de $\mathbf{M}_i$ cero
  - Establecer aleatoriamente las neuronas de la red a cero
  - Minimizar

$$\widehat{\mathcal{L}}(\theta) = -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \text{KL}\left(q_\theta(\omega)\|p(\omega)\right)$$

w.r.t. $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (conjunto de matrices)

Summary

Minimizar la divergencia entre $q_\theta(\omega)$ and $p(\omega|\mathbf{X}, \mathbf{Y})$:

- Repetir:
  - Muestrear $\widehat{\mathbf{z}}_{i,j} \sim$ Bernoulli (p_i) y hacer:

$$\widehat{\mathbf{W}}_i = \mathbf{M}_i \cdot \operatorname{diag}\left([\widehat{\mathbf{z}}_{i,j}]_{j=1}^{K_i}\right)$$

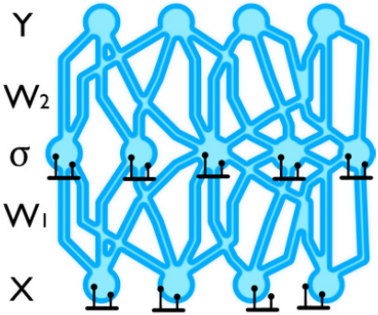
$$\widehat{\omega} = \left\{\widehat{\mathbf{W}}_i\right\}_{i=1}^{L}$$

  - = Hacer aleatoriamente columnas de $\mathbf{M}_i$ cero
  - Establecer aleatoriamente las neuronas de la red a cero
  - Minimizar

$$\widehat{\mathcal{L}}(\theta) = -\log p(\mathbf{Y}|\mathbf{X}, \widehat{\omega}) + \mathrm{KL}\left(q_\theta(\omega)||p(\omega)\right)$$

w.r.t. $\theta = \{\mathbf{M}_i\}_{i=1}^{L}$ (conjunto de matrices)

Suena familiar?



$$\widehat{\mathcal{L}}(\theta) = \overbrace{-\log p(\mathbf{Y}|\mathbf{X},\widehat{\omega})}^{=\text{ loss}} + \overbrace{\text{KL}\left(q_\theta(\boldsymbol{\omega})\|p(\boldsymbol{\omega})\right)}^{=L_2\text{ reg}}$$

1

[1] For more details see appendix of Gal and Ghahramani (2015) - `yarin.co/dropout`

¿Cómo utilizamos el *dropout* con las CNNs?

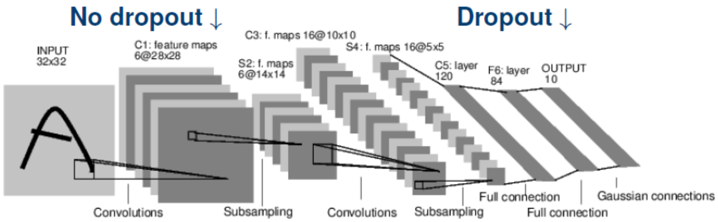


Figure: LeNet convnet structure [2]

[2] Image Source: LeCun et al. (1998)

Mejor uso del *dropout*

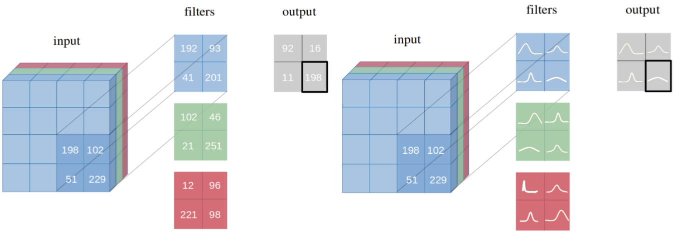
- En cambio, **media predictiva**, aprox. con integración MC:

$$\mathbb{E}_{q_\theta(\mathbf{y}^*|\mathbf{x}^*)}\left(\mathbf{y}^*\right) = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}\left(\mathbf{x}^*, \widehat{\omega}_t\right)$$

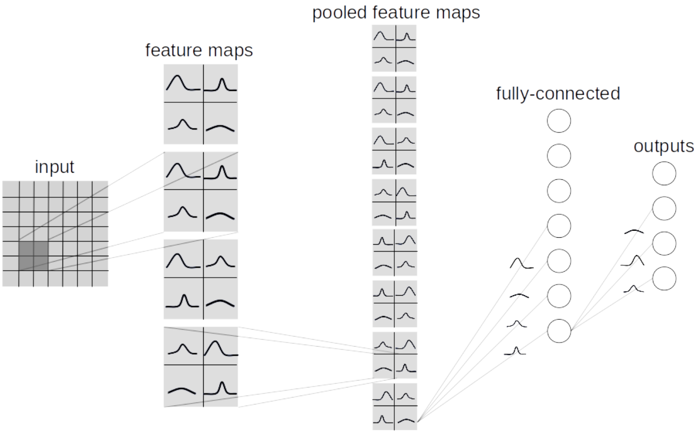
  con $\widehat{\omega}_t \sim q_\theta(\omega)$.
- En la práctica a**verage stochastic forward passes through the network** (referred to as "MC dropout").[3]
- Dropout after convolutions and averaging forward passes = **approximate inference in Bayesian convnets.**[4]

[3] Also suggested in Srivastava et al. (2014) as model averaging.
[4] See `yarin.co/bcnn` for more details

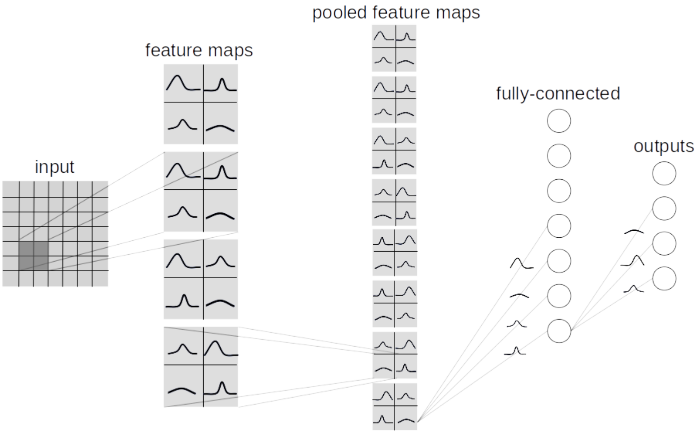
Filter weight distributions in a Bayesian Vs Frequentist approach



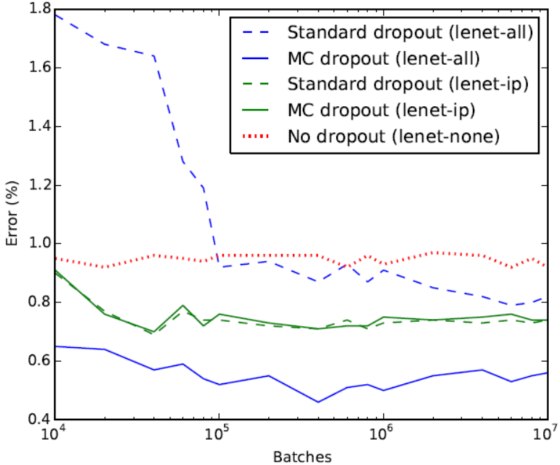
Fully Bayesian perspective of an entire CNN

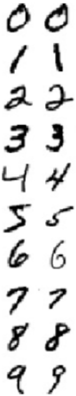


Results on MNIST and CIFAR 10 dataset



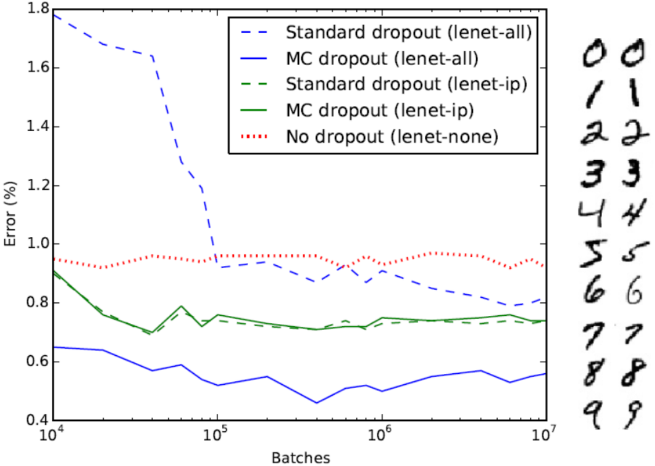
Grandes avances en MNIST I





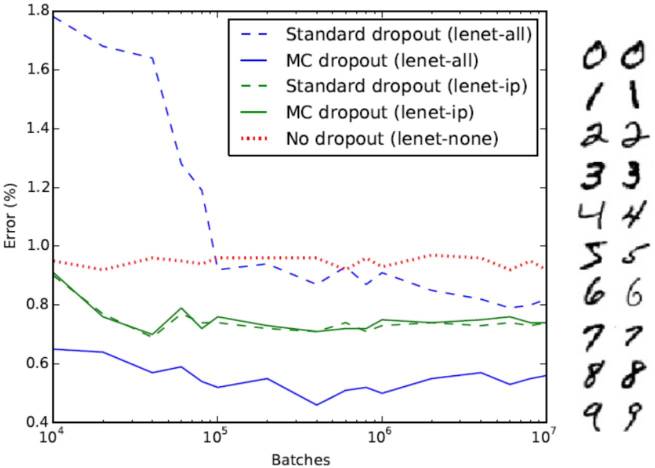
Red: standard LeNet (no dropout)

Grandes avances en MNIST II



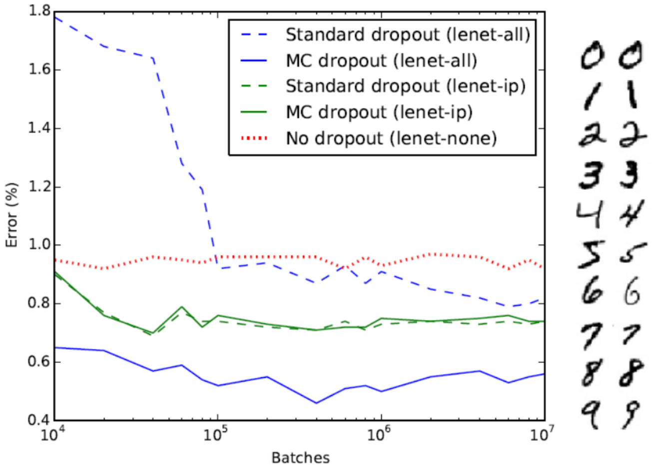
Green: standard dropout LeNet (dropout at the end)

Grandes avances en MNIST III



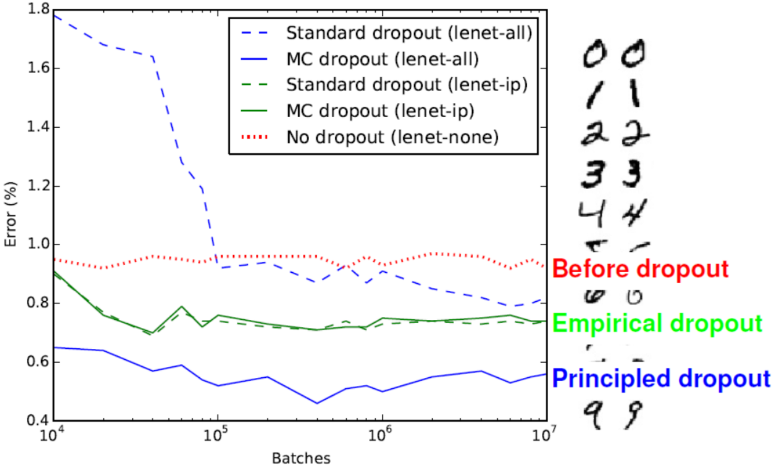
Dashed blue: Bayesian LeNet (weight averaging – FAIL)

Grandes avances en MNIST IV

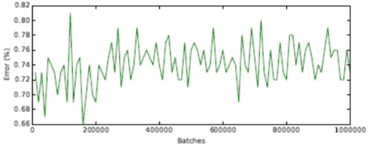


Solid blue: Bayesian LeNet (MC dropout)

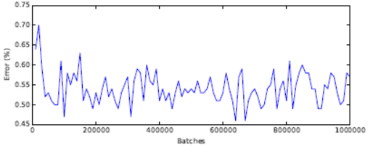
Grandes avances en MNIST V



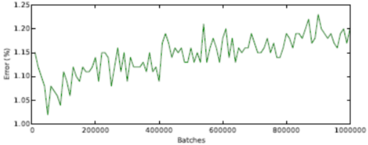
Sobre-entrenamiento en pequeñas porciones de datos



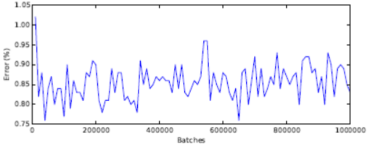
(a) Entire MNIST
Standard dropout convnet



(b) Entire MNIST
Bayesian convnet



(c) 1/4 of MNIST
Standard dropout convnet



(d) 1/4 of MNIST
Bayesian convnet

Figure: Robustness to over-fitting on smaller datasets.

State-of-the-art on CIFAR-10

	CIFAR Test Error (and Std.)	
Model	**Standard Dropout**	**MC Dropout**
NIN	10.43 (Lin et al., 2013)	$\mathbf{10.27 \pm 0.05}$
DSN	9.37 (Lee et al., 2014)	$\mathbf{9.32 \pm 0.02}$
Augmented-DSN	7.95 (Lee et al., 2014)	$\mathbf{7.71 \pm 0.09}$

Table : Bayesian techniques (MC dropout) with existing state-of-the-art

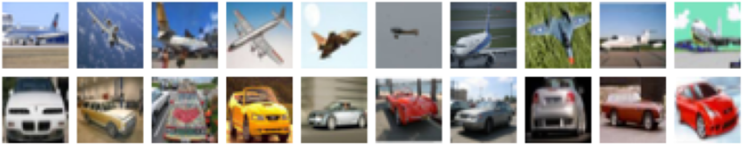
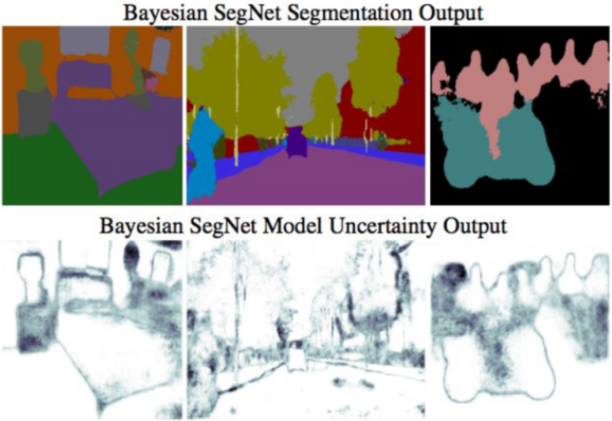


Image segmentation

- Entendimiento de la escena: ¿qué hay en una foto y dónde? (Kendall, Badrinarayanan y Cipolla, 2015)



Conclusiones

- La teoría anterior significa que el aprendizaje profundo moderno:
  - Captura procesos estocásticos subyacentes a los datos observados.
  - Puede usar vasta literatura de estadísticas Bayesianas
  - Se puede explicar mediante una teoría matemáticamente rigurosa.
  - Se puede extender de manera principista.
  - Se puede combinar con modelos/técnicas Bayesianas en un forma práctica
  - tiene estimaciones de incertidumbre incorporadas

Lo que se avecina

- Usar la teoría para responder muchas preguntas: ¿Cómo podemos ...
- ... construir modelos interpretables?
-
- ... combinar técnicas Bayesianas y modelos profundos?
- ... prácticamente se usa la incertidumbre profunda de aprendizaje en los modelos existentes?
- ... extender el aprendizaje profundo de una manera basada en principios?

Referencias I

- Denker, Schwartz, Wittner, Solla, Howard, Jackel, and Hopfield, **Large Automatic Learning, Rule Extraction, and Generalization**, Complex Systems (1987).
- Tishby, Levin, and Solla, **A statistical approach to learning and generalization in layered neural networks**, COLT (1989).
- Denker and LeCun, **Transforming neural-net output levels to probability distributions**, NIPS (1991).
- D MacKay, **A practical Bayesian framework for backpropagation networks**, Neural Computation (1992).
- GE Hinton and D van Camp, **Keeping the neural networks simple by minimizing the description length of the weights**, Computational learning theory (1993).
- R Neal, **Bayesian Learning for Neural Networks**, PhD dissertation (1995).
- D Barber and CM Bishop, **Ensemble learning in Bayesian neural networks**, Computer and Systems Sciences, (1998).

Referencias II

- A Graves, **Practical variational inference for neural networks**, NIPS (2011).
- C Blundell, J Cornebise, K Kavukcuoglu, and D Wierstra, **Weight uncertainty in neural networks**, ICML (2015).
- Krzywinski and Altman, **Points of significance: Importance of being uncertain**, Nature Methods (2013).
- Herzog and Ostwald, **Experimental biology: Sometimes Bayesian statistics are better**, Nature (2013).
- Nuzzo, **Scientific method: Statistical errors**, Nature (2014).
- Woolston, **Psychology journal bans P values**, Nature (2015).
- Ghahramani, **Probabilistic machine learning and artificial intelligence**, Nature (2015).