

Tópicos Avanzados en Entrenamiento de Redes Profundas

Dropout



Dropout

- Propuesta hacia 2012 por Hinton, Srivastava, Krizhevsky y otros colegas de Toronto, es hasta hoy una de las técnicas más populares para evitar overfitting en redes neuronales profundas.

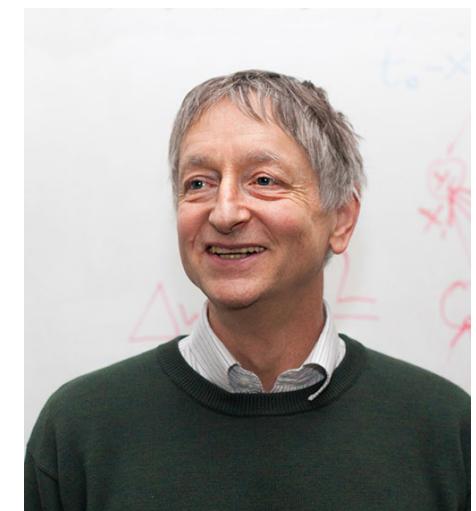
Improving neural networks by preventing co-adaptation of feature detectors

G. E. Hinton*, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov

Department of Computer Science, University of Toronto,
6 King's College Rd, Toronto, Ontario M5S 3G4, Canada

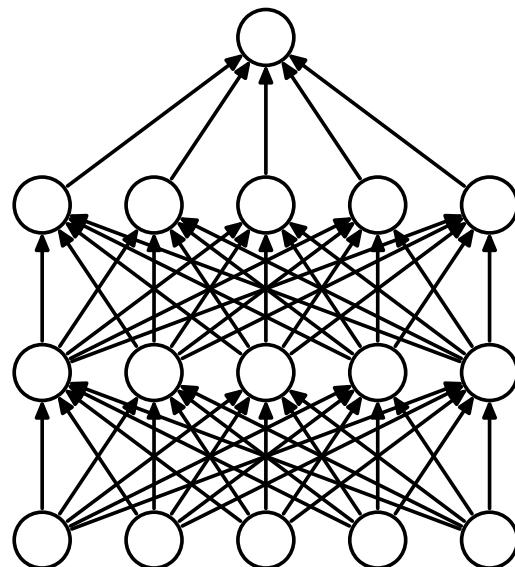
*To whom correspondence should be addressed; E-mail: hinton@cs.toronto.edu

When a large feedforward neural network is trained on a small training set, it typically performs poorly on held-out test data. This “overfitting” is greatly reduced by randomly omitting half of the feature detectors on each training case. This prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors. Instead, each neuron learns to detect a feature that is generally helpful for producing the correct answer given the combinatorially large variety of internal contexts in which it must operate. Random “dropout” gives big improvements on many benchmark tasks and sets new records for speech and object recognition.

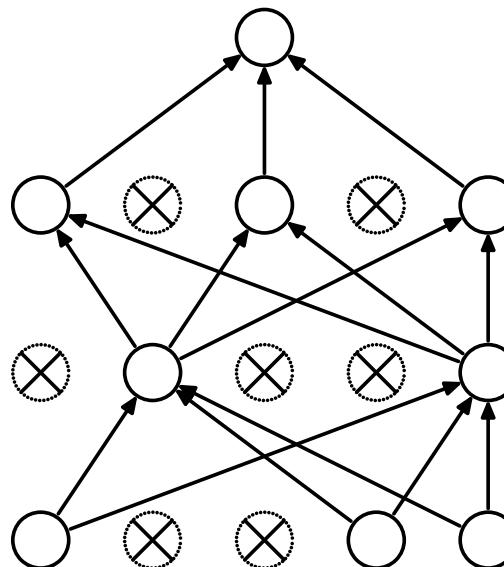


Idea

- Durante cada iteración de entrenamiento, mantener cada neurona de la red con probabilidad p y ocultarla con probabilidad $1-p$.



(a) Standard Neural Net



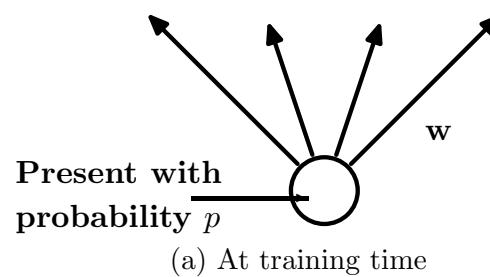
(b) After applying dropout.

* En la práctica, se recomienda $p = 0.5$ ó $p \in [0.5, 0.8]$.

Entrenamiento

- En el caso de una capa densa con S unidades, input x y output y , podemos formalizar esta idea definiendo una máscara estocástica $m \in \{0,1\}^S$ con $m_i = \sim \text{Ber}(p)$ y actualizando la ecuación de la capa como sigue:

$$y = m \odot g(Wx - b)$$

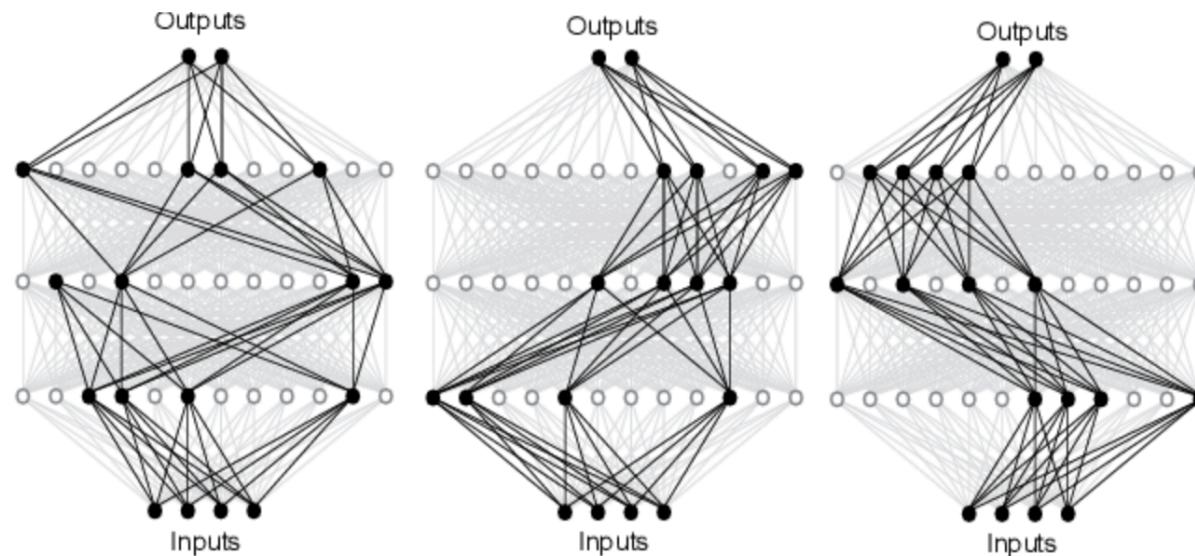


- La máscara se muestrea en cada iteración de entrenamiento (*) de manera independiente de las otras neuronas de la red.

(*) normalmente se usa una máscara diferente para cada mini-batch y cada ejemplo dentro de un mini-batch.

Predictión

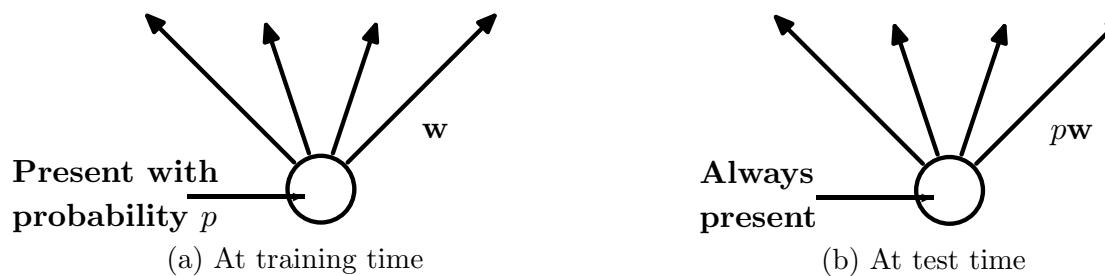
- Una vez entrenada la red, ¿qué máscara utilizamos?
- Una posibilidad sería ejecutar muchas veces la predicción y promediar (o tomar una estadística de) los resultados.



Predicción

- Lo usual es que una vez entrenada la red, la capa se opere sin la máscara, pero multiplicando las activaciones por p . Esto es equivalente a tomar el valor esperado de y

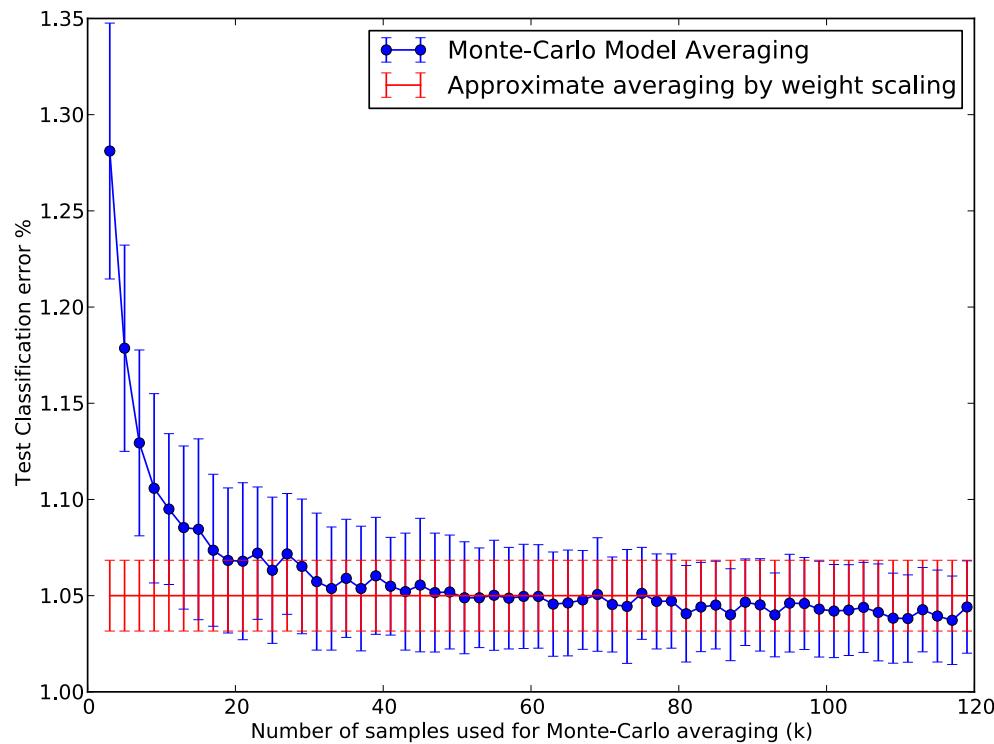
$$\mathbb{E}[y] = \mathbb{E}[m \odot g(Wx - b)] = p \cdot g(Wx - b)$$



- Claramente, esto es equivalente a corregir los pesos salientes en vez de las activaciones. En la práctica se suele multiplicar y por $1/(1-p)$ durante el *entrenamiento* en un procedimiento a veces llamado *Inverse Dropout*.

Predicción

- Se ha mostrado que cualquiera de estos métodos es una buena aproximación del método estocástico.

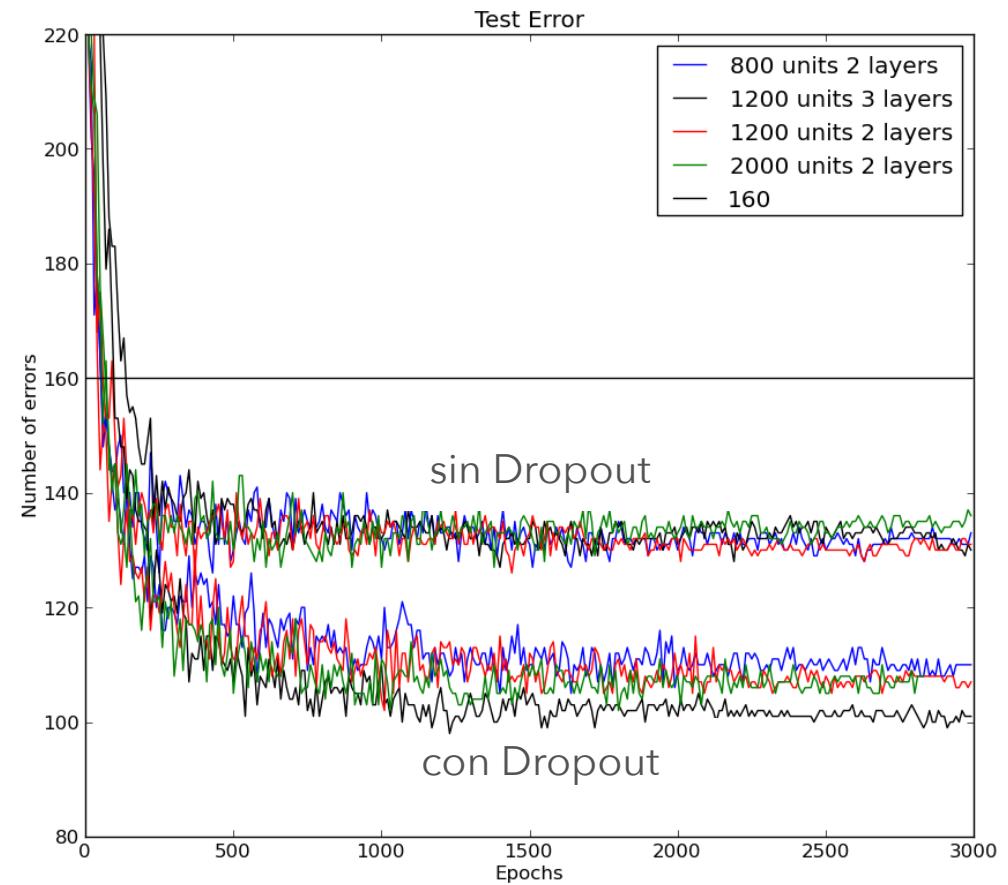


* Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

Efecto sobre el Error de Predicción

- Algunos de los resultados originales (Hinton et al. 2012)

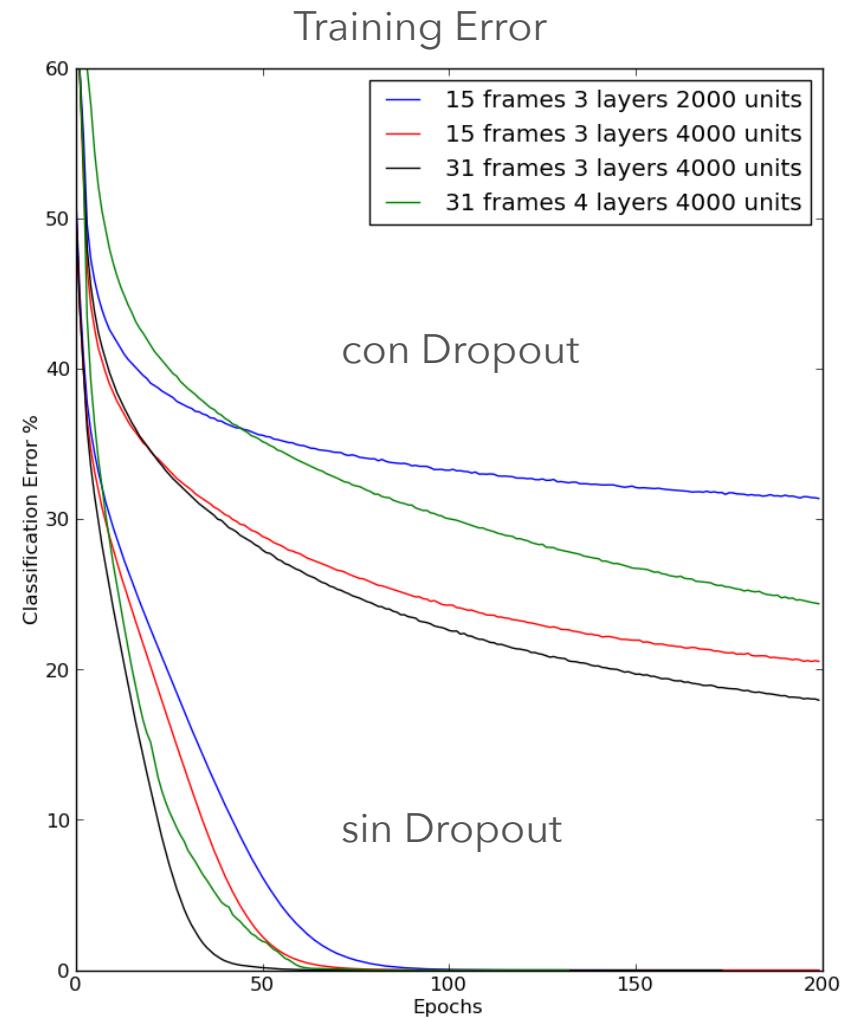
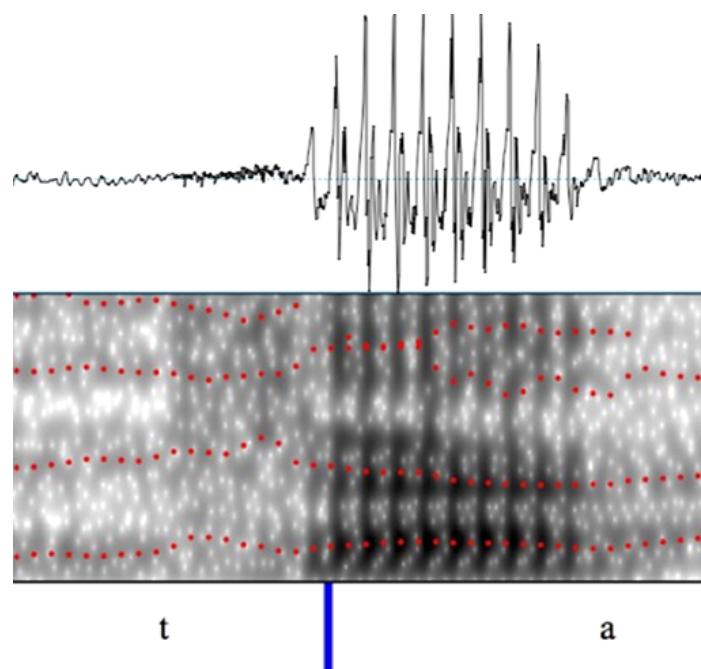
MNIST



Efecto sobre el Error de Predicción

- Algunos de los resultados originales (Hinton et al. 2012)

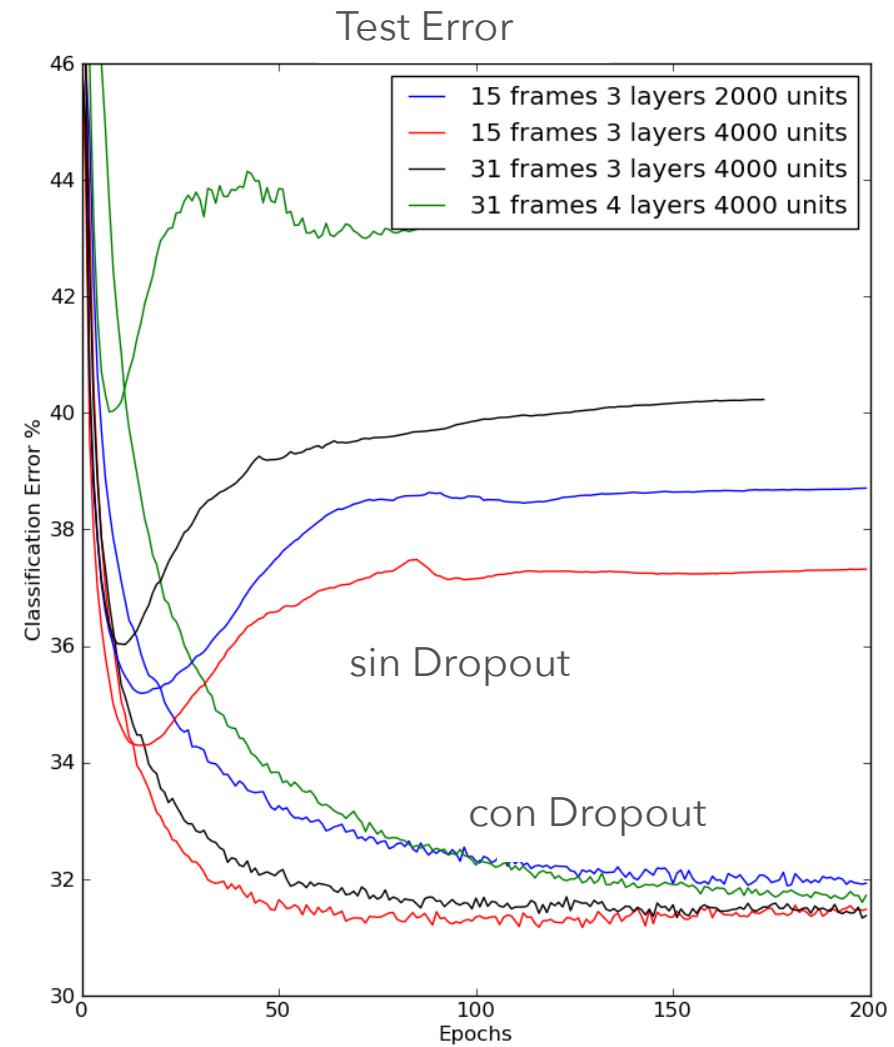
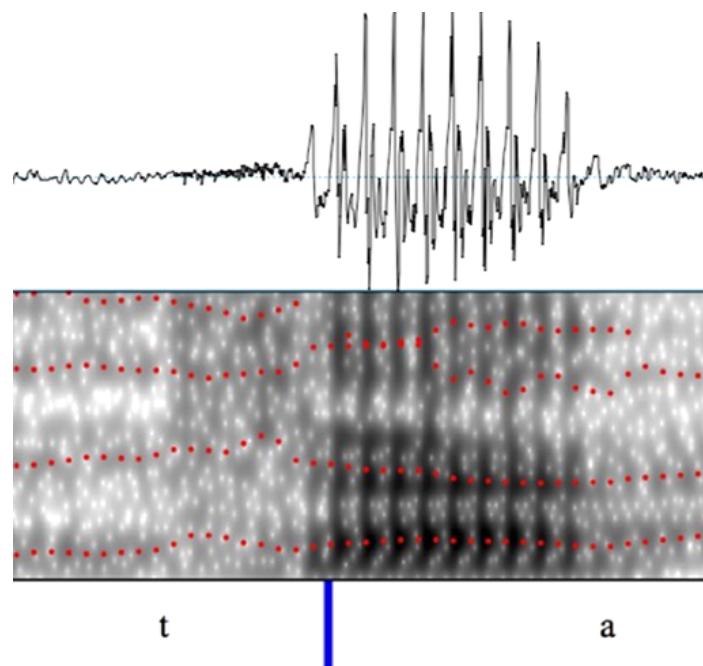
TIMIT



Efecto sobre el Error de Predicción

- Algunos de los resultados originales (Hinton et al. 2012)

TIMIT

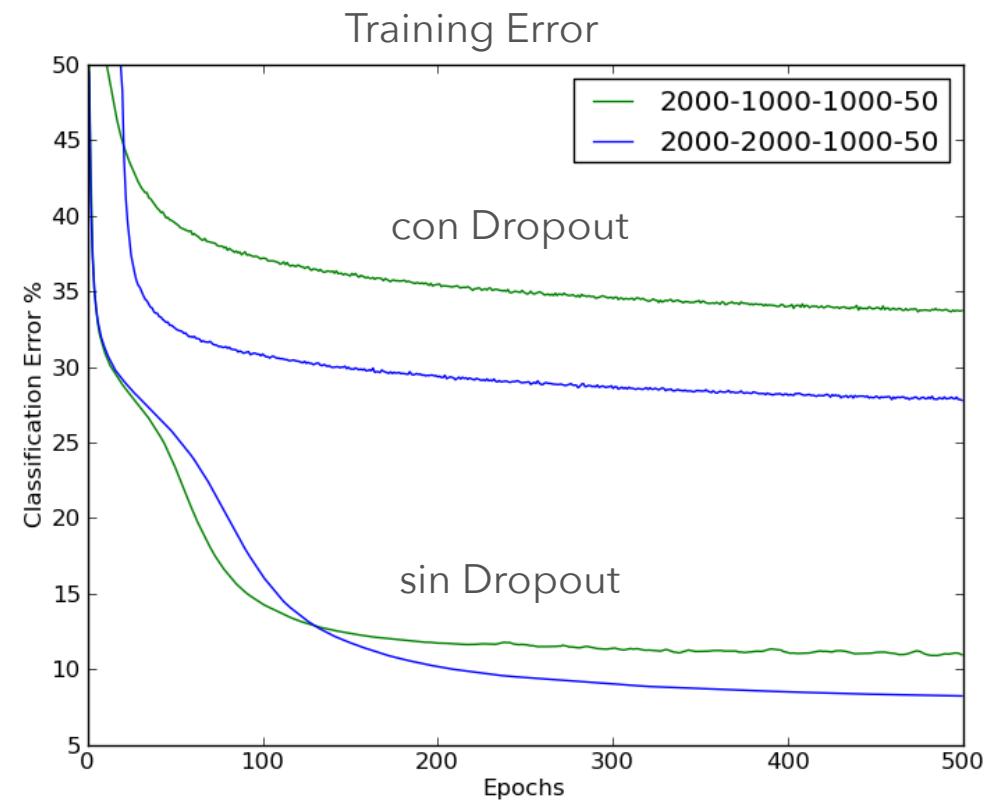


Efecto sobre el Error de Predicción

- Algunos de los resultados originales (Hinton et al. 2012)

REUTERS

```
<REUTERS TOPICS='YES' LEWISPLIT='TRAIN'  
CGISPLIT='TRAINING-SET' OLDDID='12981' NEWID='798'>  
<DATE> 2-MAR-1987 16:51:43.42</DATE>  
<TOPICS><D>livestock</D><D>hog</D></TOPICS>  
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>  
<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork  
Congress kicks off tomorrow, March 3, in Indianapolis with 160  
of the nations pork producers from 44 member states determining  
industry positions on a number of issues, according to the  
National Pork Producers Council, NPPC.  
Delegates to the three day Congress will be considering 26  
resolutions concerning various issues, including the future  
direction of farm policy and the tax law as it applies to the  
agriculture sector. The delegates will also debate whether to  
endorse concepts of a national PRV (pseudorabies virus) control  
and eradication program, the NPPC said. A large  
trade show, in conjunction with the congress, will feature  
the latest in technology in all areas of the industry, the NPPC  
added. Reuter  
&lt;></BODY></TEXT></REUTERS>
```

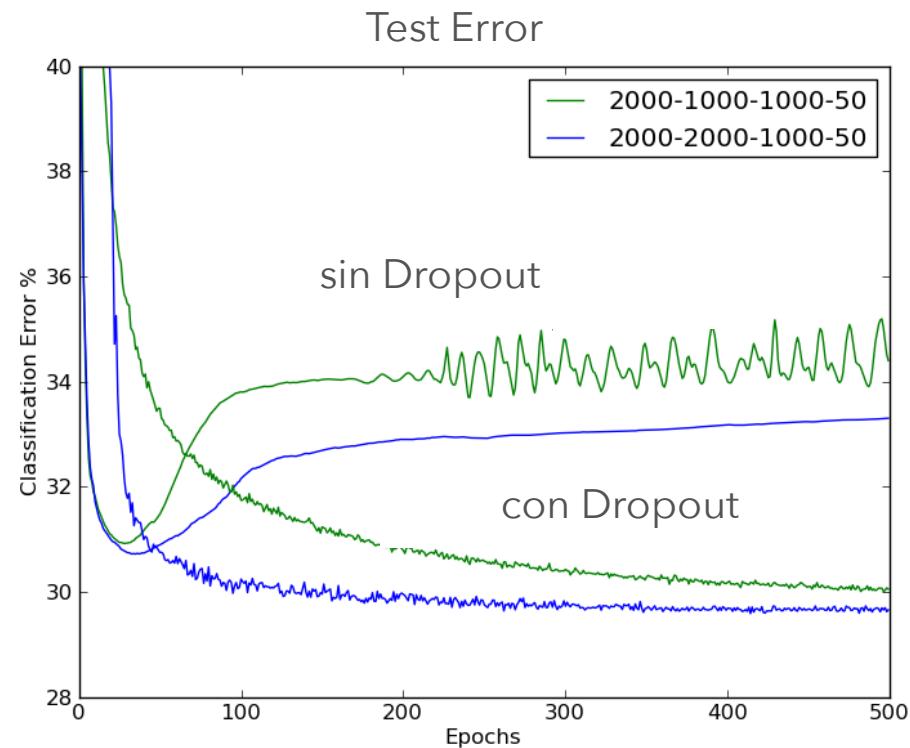


Efecto sobre el Error de Predicción

- Algunos de los resultados originales (Hinton et al. 2012)

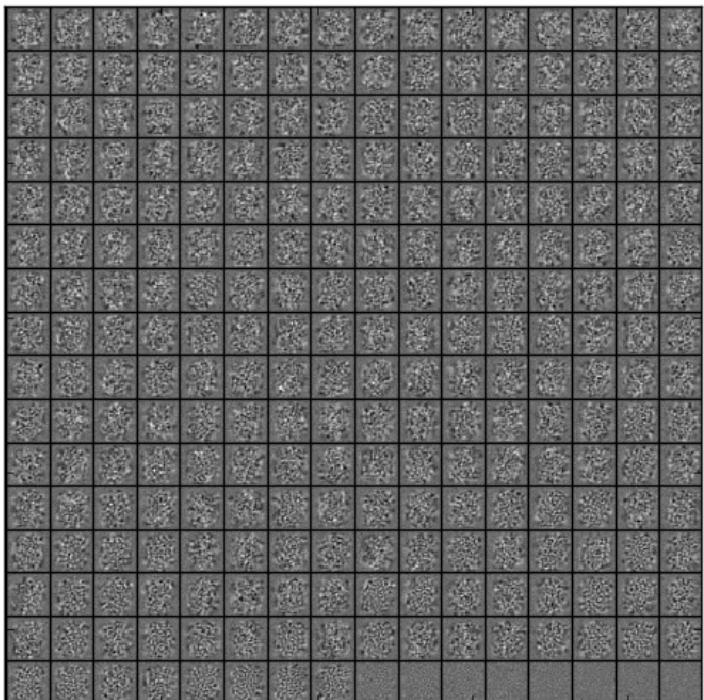
REUTERS

```
<REUTERS TOPICS='YES' LEWISPLIT='TRAIN'  
CGISPLIT='TRAINING-SET' OL DID='12981' NEWID='798'>  
<DATE> 2-MAR-1987 16:51:43.42</DATE>  
<TOPICS><D>livestock</D><D>hog</D></TOPICS>  
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>  
<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork  
Congress kicks off tomorrow, March 3, in Indianapolis with 160  
of the nations pork producers from 44 member states determining  
industry positions on a number of issues, according to the  
National Pork Producers Council, NPPC.  
Delegates to the three day Congress will be considering 26  
resolutions concerning various issues, including the future  
direction of farm policy and the tax law as it applies to the  
agriculture sector. The delegates will also debate whether to  
endorse concepts of a national PRV (pseudorabies virus) control  
and eradication program, the NPPC said. A large  
trade show, in conjunction with the congress, will feature  
the latest in technology in all areas of the industry, the NPPC  
added. Reuter  
&lt;&gt;</BODY></TEXT></REUTERS>
```

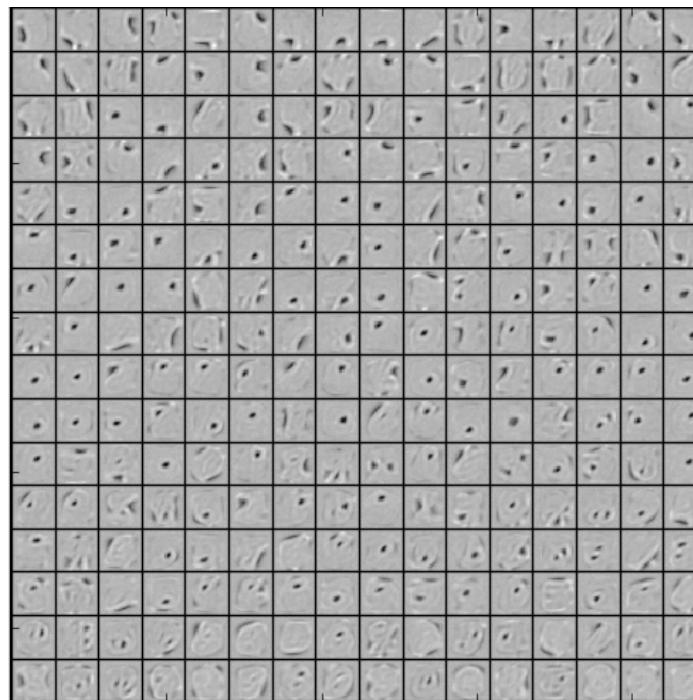


Efecto sobre los Pesos

- Experimentalmente, se observa también que Dropout permite obtener discriminantes más diferenciados y “ralos” en las capas ocultas.



(a) Without dropout

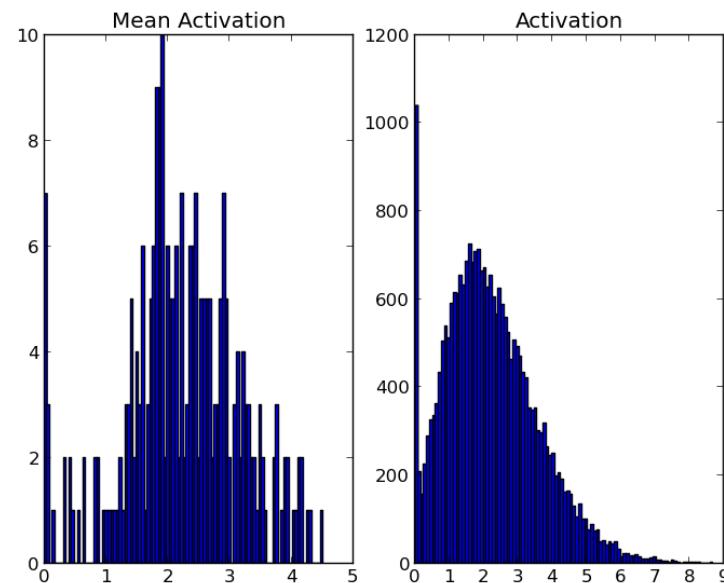


(b) Dropout with $p = 0.5$.

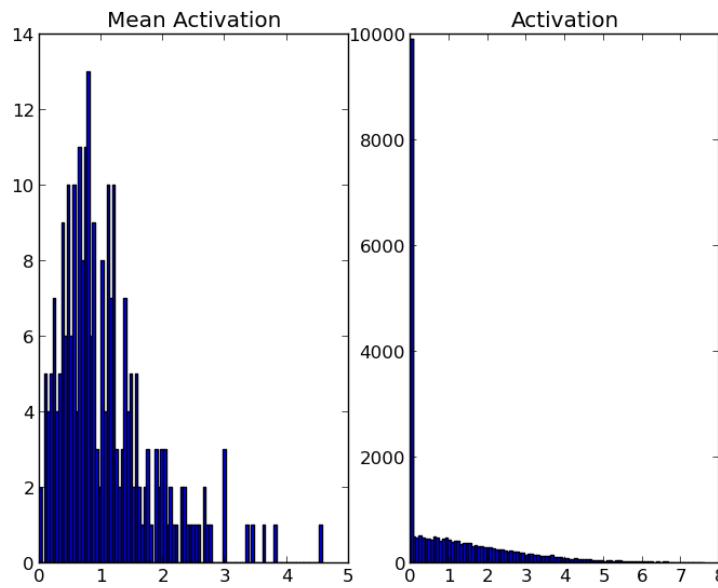
(AE entrenado en MNIST)

Efecto sobre las Activaciones

- Las activaciones de las capas ocultas son también más ralas (sparse) reduciendo la dimensionalidad efectiva de las representaciones aprendidas por la red.



(a) Without dropout



(b) Dropout with $p = 0.5$.

(AE entrenado en MNIST)

Co-Adaptación

- La hipótesis original de los autores es que Dropout reduce el fenómeno de co-adaptación de neuronas en la red (*) y por lo tanto la robustez del modelo a la ausencia de ciertas características específicas de los datos de entrada que no se observan en todos los datos de pruebas.
- Intuitivamente: si un atributo sólo es útil cuando hay otros presentes, huele a overfitting.
- Esta interpretación nos hace recordar el viejo método de entrenar una red neuronal “con ruido”

Ejemplo original

$$(x, y)$$

Ejemplo “corrupto”

$$(x + \epsilon, y)$$

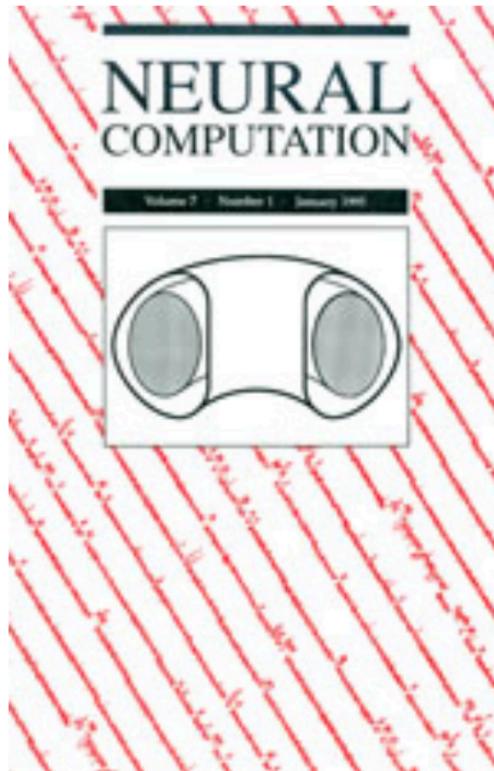
$$\mathbb{E}(\epsilon) = 0$$

$$\mathbb{E}(\epsilon\epsilon^T)_{ij} = \delta_{ij}\gamma^2$$

(*) Dos unidades están co-adaptadas si sus activaciones están fuertemente correlacionadas.

Training with Noise

- De acuerdo a esta interpretación, Dropout se podría considerar una generalización en profundidad del viejo método de "entrenar con ruido"



Training with Noise is Equivalent to Tikhonov Regularization

Chris M. Bishop

Posted Online April 04, 2008

<https://doi.org/10.1162/neco.1995.7.1.108>

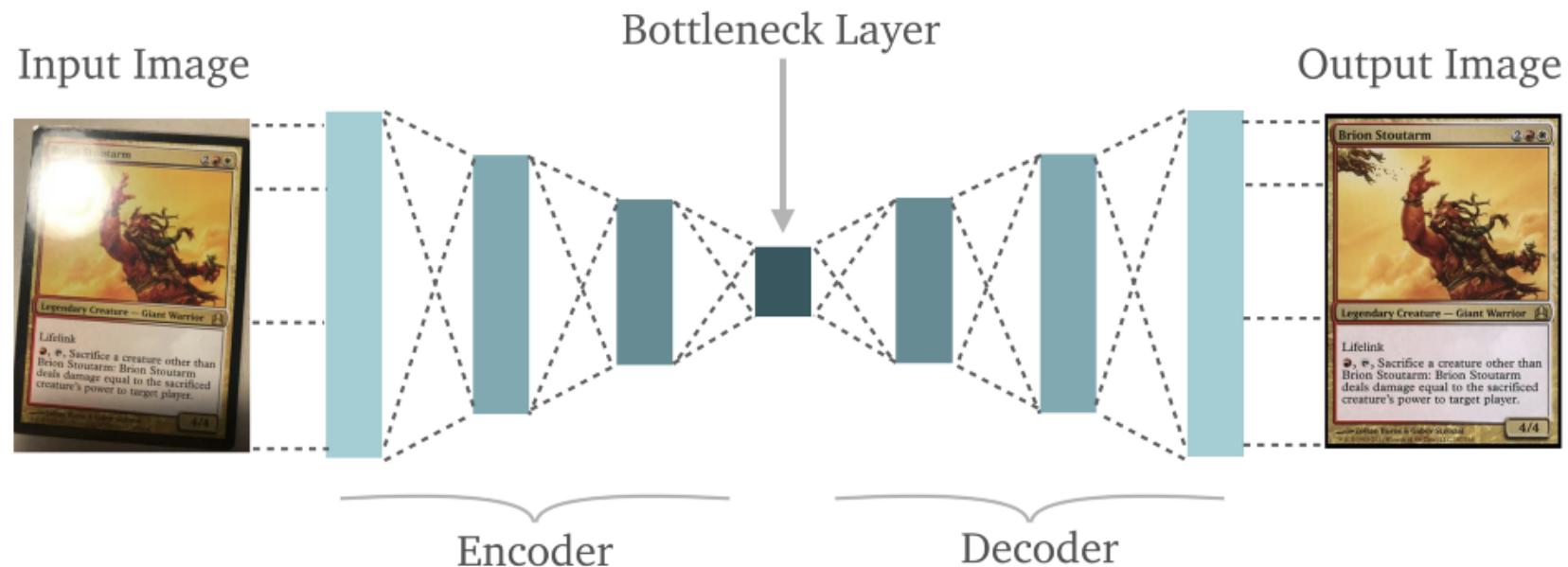
© 1995 Massachusetts Institute of Technology

Neural Computation
Volume 7 | Issue 1 | January 1995
p.108-116



Training with Noise

- La máscara Bernoulli (salt & pepper) no es común problemas con atributos de alto nivel, pero si en imágenes. Por ejemplo, los denoising auto-encoders aplicados a imágenes normalmente usan este tipo de máscara.



Gaussian Dropout

- Hacia 2014, Srivastava, Hinton y otros de entre los autores originales de Dropout muestran que sustuir la máscara Bernoulli por una máscara (multiplicativa) Gaussiana M con $M_{ij} \sim \mathcal{N}(1, p(1 - p)/p)$ igualmente efectivo y en ocasiones mejor.
- Este resultado fortalece los links con el viejo método de regularización y motiva una serie de variantes de Dropout centradas en perturbaciones continuas en vez de binarias/discretas, como el Dropout variacional de Kingma et al. 2015.

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

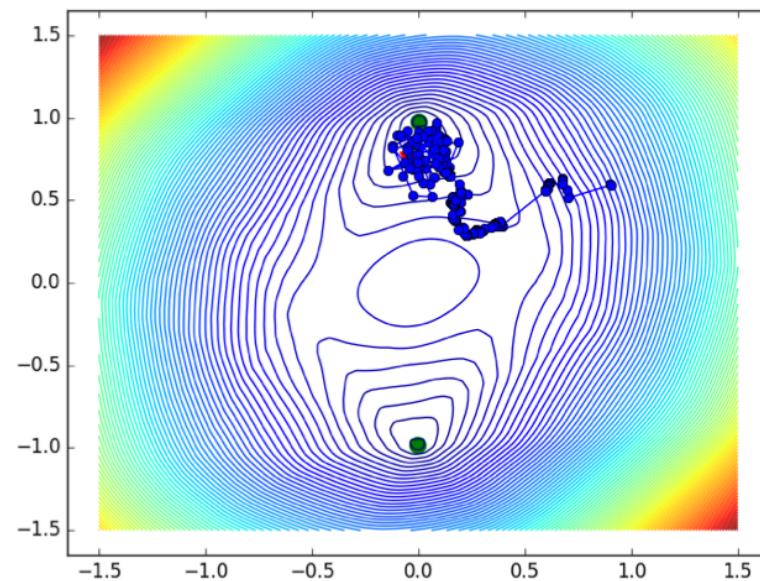
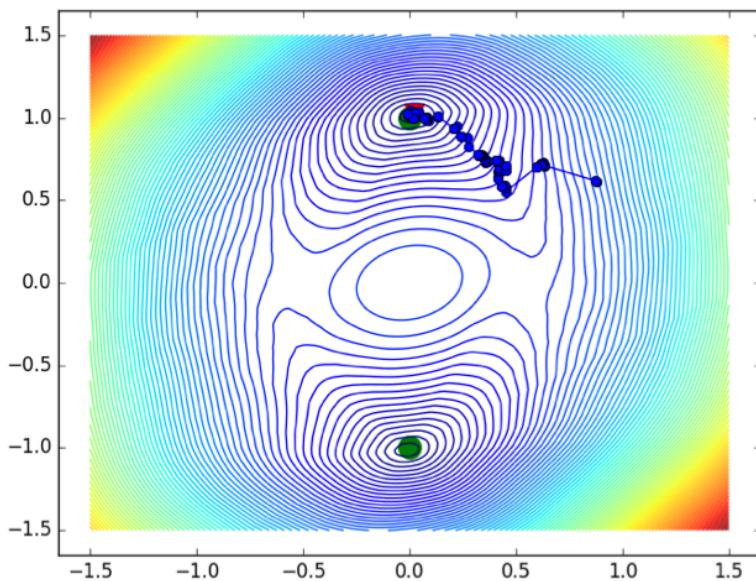
Kingma, Durk P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." *Advances in neural information processing systems*. 2015.

(*) Dropout con escalamiento durante el entrenamiento, toma el valor de la activación multiplicado por $1/p$ con probabilidad p , lo que resta equivalente a multiplicar por una Bernoulli escalada.



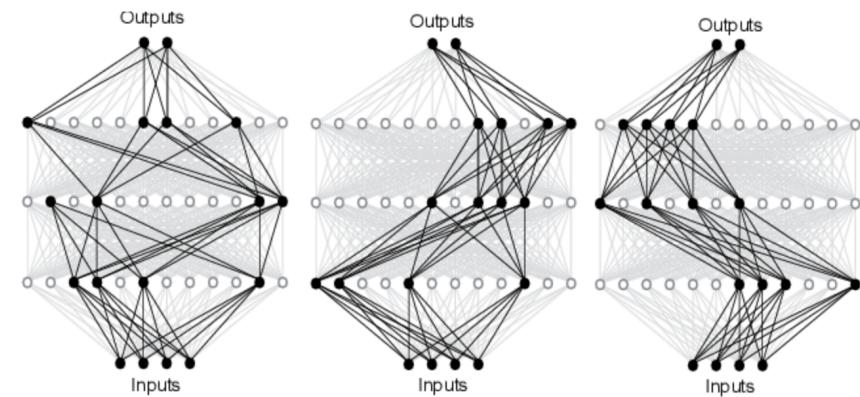
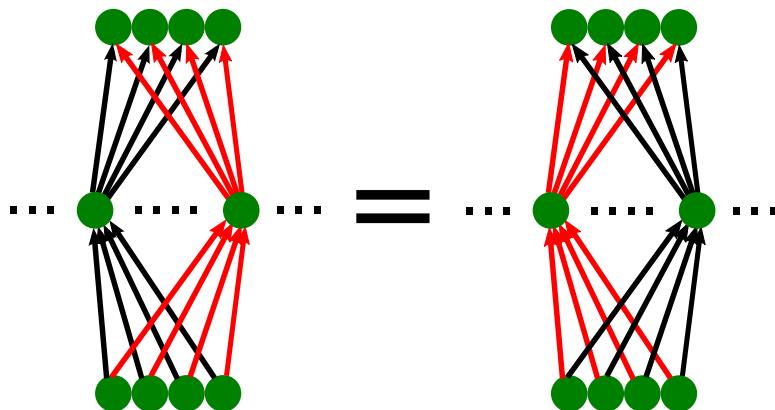
Co-Adaptación

- Notemos también que, como las diferentes unidades de la red aplicarán máscaras independientes, no todas las unidades de una capa estarán activas en una determinada iteración de entrenamiento. De este modo, la secuencia de ajustes de estas unidades durante el entrenamiento varía.



Co-Adaptación

- Esto reduce la probabilidad de que dos unidades de la misma capa terminen aprendiendo la misma tarea/atributo, aún si parten de estados similares.



Ensemble vía Dropout

- Otra interpretación (más comúnmente mencionada en la literatura) es que Dropout permite aproximar un *ensemble de modelos* en la misma arquitectura de red.



- En machine learning, un *ensemble* construye un modelo combinando múltiples modelos sencillos. En particular *Bagging* genera una predicción continua promediando K regresores que se entrena para ser aproximadamente independientes:

$$f(x) = \frac{1}{K} \sum_i f_i(x)$$

Bagging

- Un Bagging “ideal” (predictores realmente independientes) logra reducir el error de predicción reduciendo la varianza del predictor:

$$\begin{aligned}\mathbb{E} [(y - f_S(x))^2] &= \mathbb{E} [f_S(x)^2] - 2\mathbb{E} [f_S(x)] \mathbb{E} [y] + \mathbb{E} [y^2] \\&= \mathbb{V} [f_S(x)] + \mathbb{E} [f_S(x)]^2 - 2\mathbb{E} [f_S(x)] f(x) + \mathbb{V} [y] + \mathbb{E} [y]^2 \\&= \mathbb{V} [f_S(x)] + \mathbb{E} [f_S(x)]^2 - 2\mathbb{E} [f_S(x)] f(x) + \mathbb{V} [y] + f(x)^2 \\&= \mathbb{V} [f_S(x)] + (\mathbb{E} [f_S(x)] - f(x))^2 + \mathbb{V} [y] \\&= \mathbb{V} [f_S(x)] + (\mathbb{E} [f_S(x)] - f(x))^2 + \sigma^2\end{aligned}$$

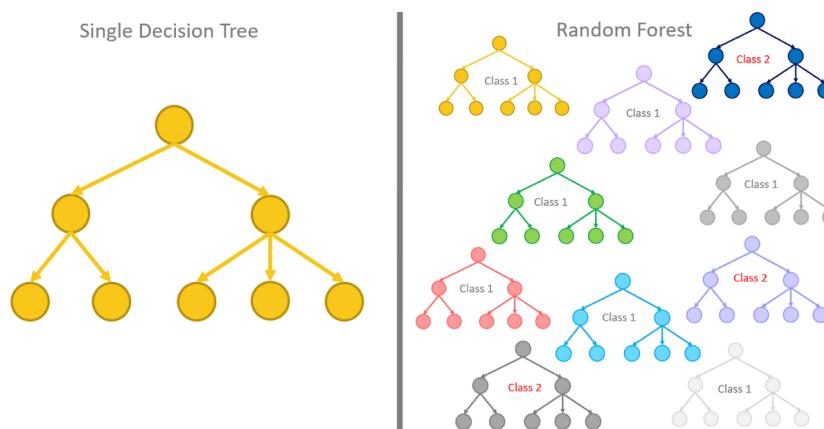
error de predicción = sesgo + varianza + varianza intrínseca de y

Bagging

- Un Bagging o RF “ideal” (predictores realmente independientes) logra reducir el error de predicción reduciendo la varianza del predictor y manteniendo el sesgo constante.

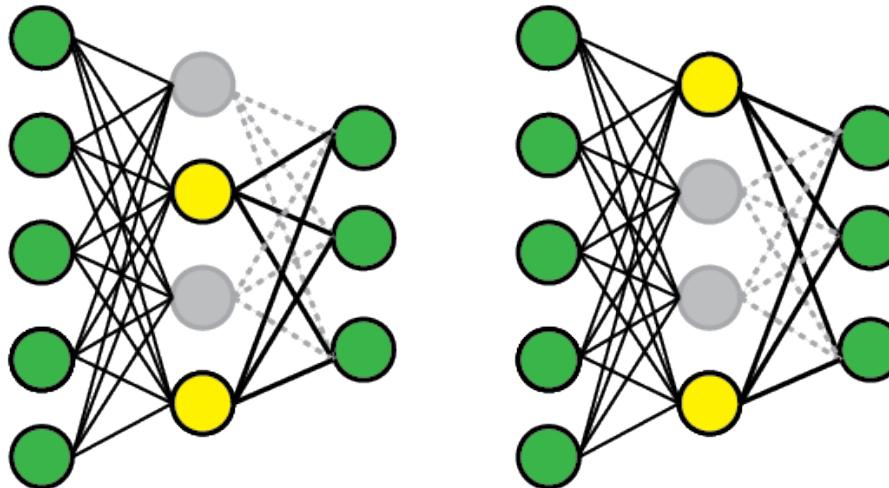
$$\text{Bias}(f) = \frac{1}{K} \sum_i \text{Bias}\left(f^{(i)}(x)\right) = B$$

$$\mathbb{V}(f) = \frac{1}{K^2} \sum_i \mathbb{V}\left(f^{(i)}(x)\right) = \frac{C}{K}$$



Ensemble vía Dropout

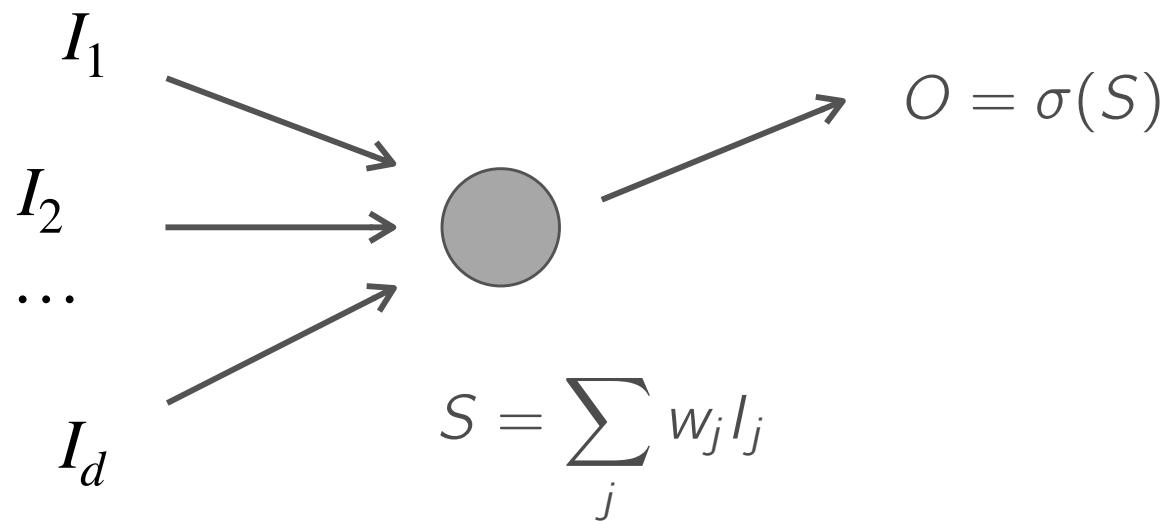
- Durante el entrenamiento, la máscara hace que sea *una sub-red* la que está siendo entrenada (en cada iteración y para ejemplo). ¿Cuántas subredes posibles existen?



- Al tomar valor esperado de la salida de cada capa estamos aproximando la media de un número exponencialmente grande de sub-modelos.

Ensemble vía Dropout

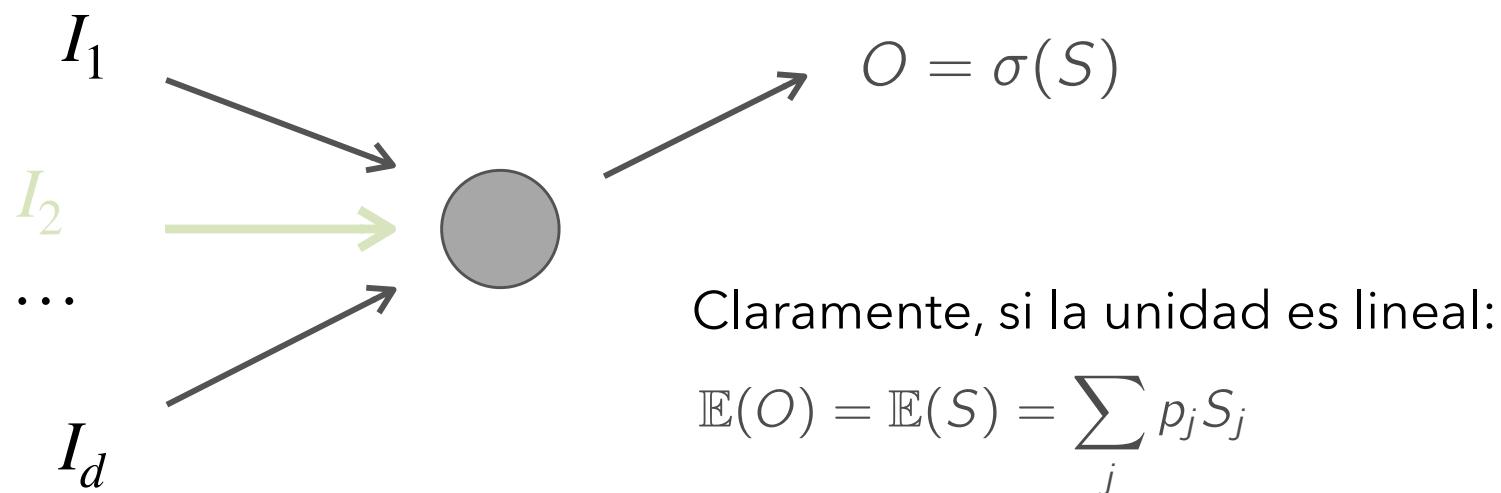
- Baldi et al. 2013 formaliza matemáticamente algunas de estas ideas. Consideremos la dinámica de una neurona de salida durante el entrenamiento:



Baldi, Pierre, and Peter J. Sadowski. "Understanding Dropout." *Advances in Neural Information Processing Systems*. 2013.

Ensemble vía Dropout

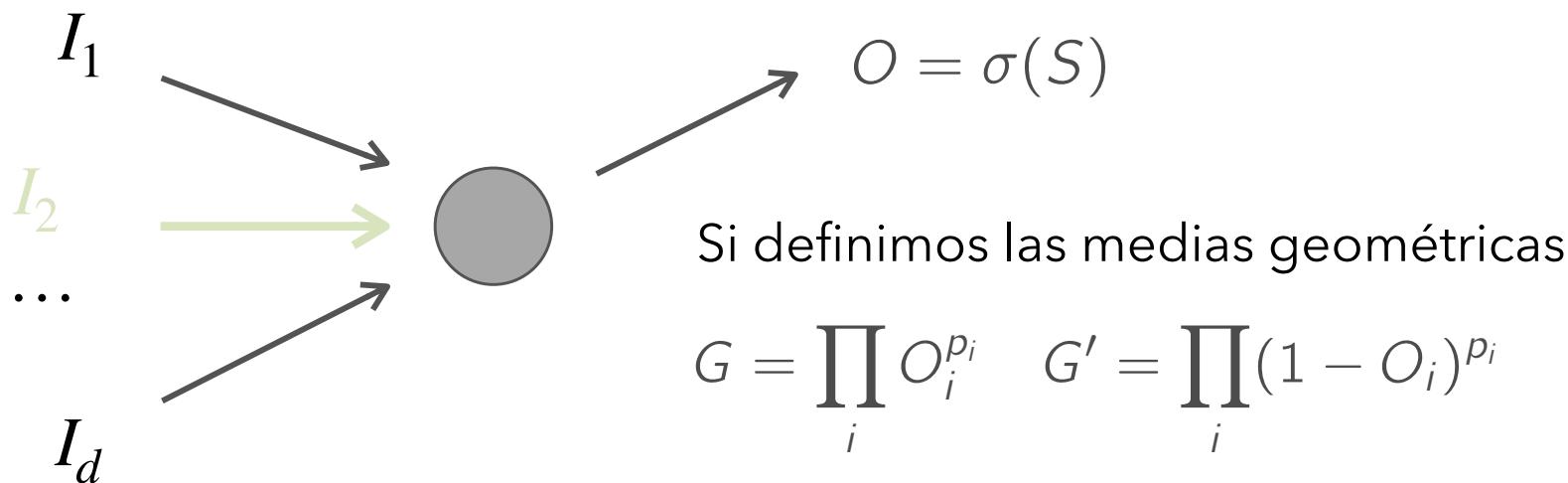
- Debido a Dropout, los inputs que entran a la unidad dependerán de las sub-redes incidentes que se realizan en una determinada iteración. Si denotamos por S_1, S_2, \dots, S_m las distintas posibles sumas (pre-activaciones) y por p_1, p_2, \dots, p_m sus probabilidades, ¿Podemos calcular el valor esperado de O?



de modo que el valor esperado de la respuesta es un ensemble de las salidas de las posibles sub-redes.

Ensemble vía Dropout

- Baldi muestra que en el caso de neuronas sigmoidales esto sigue siendo aproximadamente cierto:



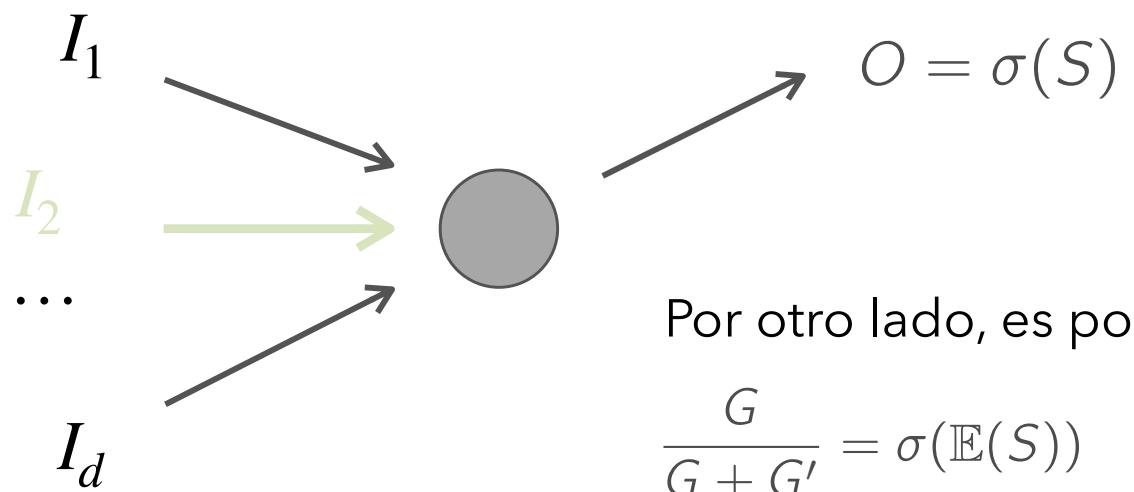
Usando una expansión en series de Taylor de orden 2, obtenemos por un lado que:

$$\left| \mathbb{E}(O) - \frac{G}{G + G'} \right| \approx \frac{V|1 - 2\mathbb{E}(O)|}{1 - 2V} \leq 2\mathbb{E}(O)(1 - \mathbb{E}(O))|1 - 2\mathbb{E}(O)|$$

media geométrica normalizada $\text{Var}(O)$

Ensemble vía Dropout

- Baldi muestra que en el caso de neuronas sigmoidales esto sigue siendo aproximadamente cierto:



Por otro lado, es posible demostrar que

$$\frac{G}{G + G'} = \sigma(\mathbb{E}(S))$$

De modo que:

$$\mathbb{E}(O) \approx \frac{G}{G + G'} = \sigma(\mathbb{E}(S))$$

Es decir, la unidad sigue aproximando el valor esperado de la salida de las sub-redes implícitamente entrenadas durante el entrenamiento.

Ensemble vía Dropout

- Baldi analiza también el gradiente que recibiría una neurona sigmoidal entrenada con la loss cross entropy usando *Dropout*.

$$E_{\text{ens}}(t) = - (t \log \mathbb{E}(S) + (1 - t)(1 - \log \mathbb{E}(S)))$$

$$E_{\text{drop}}(t) = - (t \log O + (1 - t) \log(1 - O))$$

$$\mathbb{E} \left(\frac{\partial E_{\text{drop}}}{\partial w_j} \right) \approx \frac{\partial E_{\text{ens}}}{\partial w_j} + \sigma'(\mathbb{E}(S)) I_j^2 \text{Var}(M_j) w_j$$

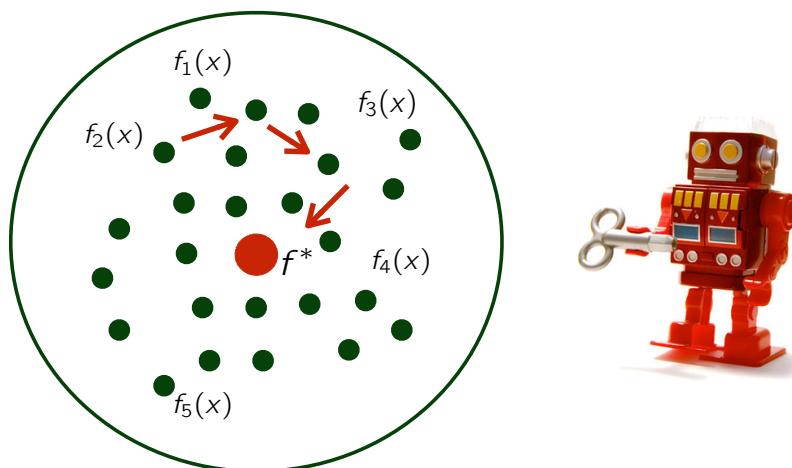
Es decir, con Dropout el gradiente es aproximadamente el que recibiría un ensemble regularizado con *Weight Decay* (L2 penalty).

El mayor penalty (mayor regularización) se obtiene usando una máscara Dropout (M) con máxima varianza, i.e., $p=1/2$.

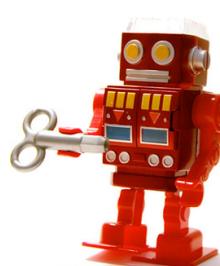
Dropout como Método Bayesiano

- En línea con esta interpretación, se ha planteado que Dropout podría ser una aproximación al tipo de predicciones que utilizan los métodos Bayesianos

$$p(y|x) = \int_H p(y|x, h)dP(h)$$



Learning as Search

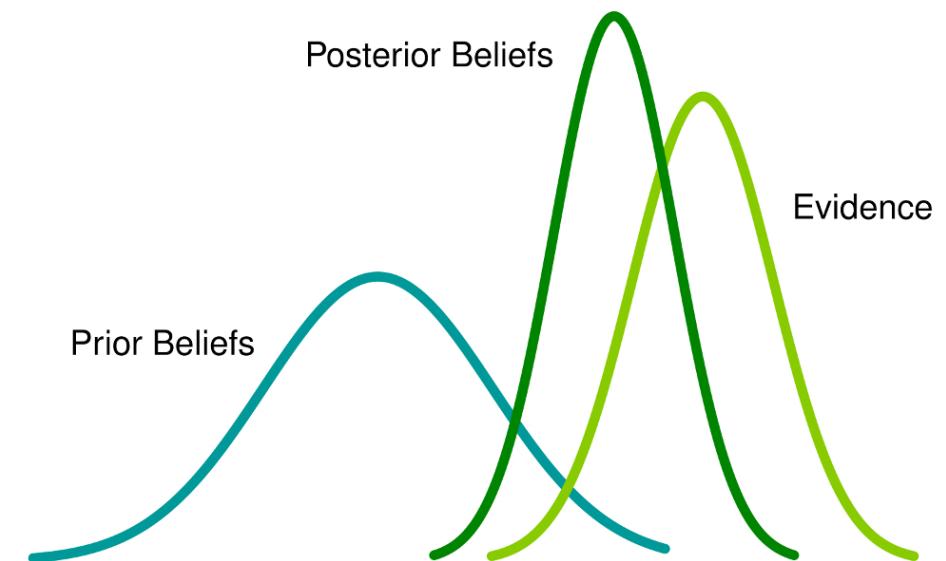


Dropout como Método Bayesiano

- En línea con esta interpretación, se ha planteado que Dropout podría ser una aproximación al tipo de predicciones que utilizan los métodos Bayesianos

$$p(y|x, D) = \int p(y|x, D, \theta)p(\theta|D)d\theta$$

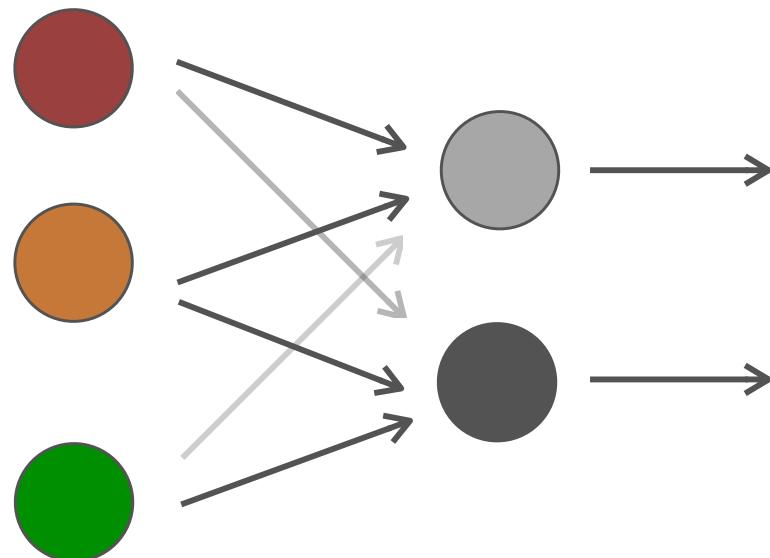
$$p(\theta|D) \propto p(D|\theta)p(\theta)$$



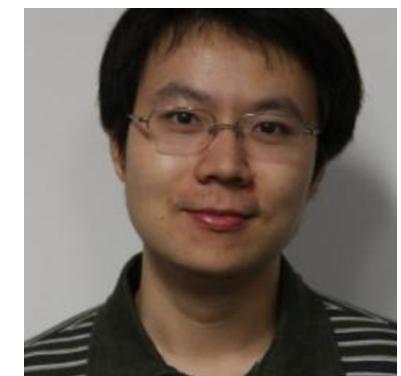
DropConnect

- Hacia 2013, Li Wan et al. proponen una de las variantes más estudiadas de Dropout. En vez ocultar una neurona, los autores proponen ocultar los pesos de una capa con probabilidad p . Formalmente, tenemos

$$y = g(M \odot W x - m \odot b) \quad M_{ij} \sim \text{Ber}(p), \quad m_i \sim \text{Ber}(p)$$



Wan, Li, et al. "Regularization of Neural Networks using Dropconnect."
International Conference on Machine Learning. 2013.



DropConnect

- Una vez concluido el entrenamiento, en vez de escalar los pesos, los autores proponen aproximar la pre-activación de la neurona usando el *Teorema del Límite Central*. Formalmente, para la unidad i de una capa

$$M \odot Wx = \sum_j (W_{:j}x_j)m_{:j} \approx \mathcal{N}(pWx, \Sigma_x)$$
$$\Sigma_x = p(1 - p)(W \odot W)(x \odot x)$$

Suma pesada de Bernoulli's
 $(M \odot Wx)_i = \sum_j (W_{ij}x_j)m_{ij}$

- La activación de la neurona se obtiene muestreando de esta Gaussiana y pasando el resultado por la no linealidad correspondiente a la capa.
- En ocasiones, DropConnect (a veces llamado weight Dropout) puede regularizar redes complejas más eficazmente que la versión estándar.
- Una vez entrenada la red, se sigue aplicando Dropout de manera estocástica en vez de ajustar los pesos o activaciones por el inverso de p .

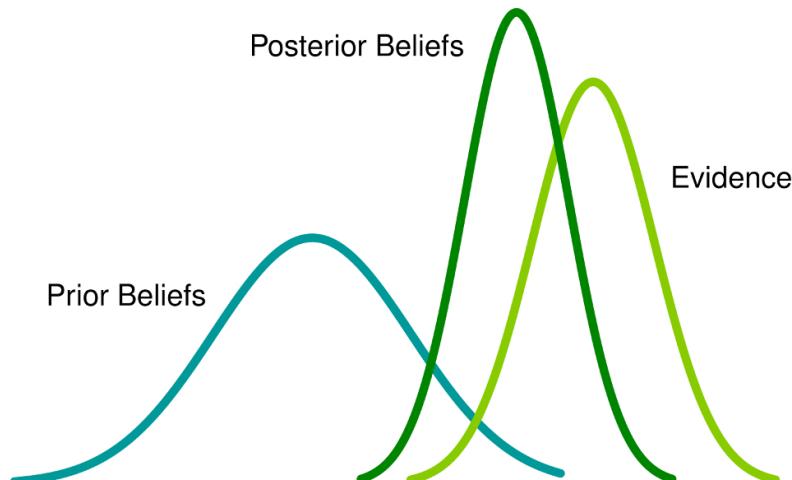
Dropout Variacional

- No es difícil demostrar que aplicar una máscara multiplicativa Gaussiana $m_{ij} \sim \mathcal{N}(1,\alpha)$ a las activaciones resulta equivalente a aplicarla a los pesos, es decir, elegir los pesos de una unidad como $w_i = \theta_i s_i$ con $s_i \sim \mathcal{N}(1,\alpha)$.
- Kingma et al. notan en 2015 que esto se puede interpretar como una elección del a-posteriori en un método Bayesiano.

$$p(y|x, D) = \int p(y|x, D, \theta)p(\theta|D)d\theta$$

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

Kingma, Durk P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." *Advances in neural information processing systems*. 2015.



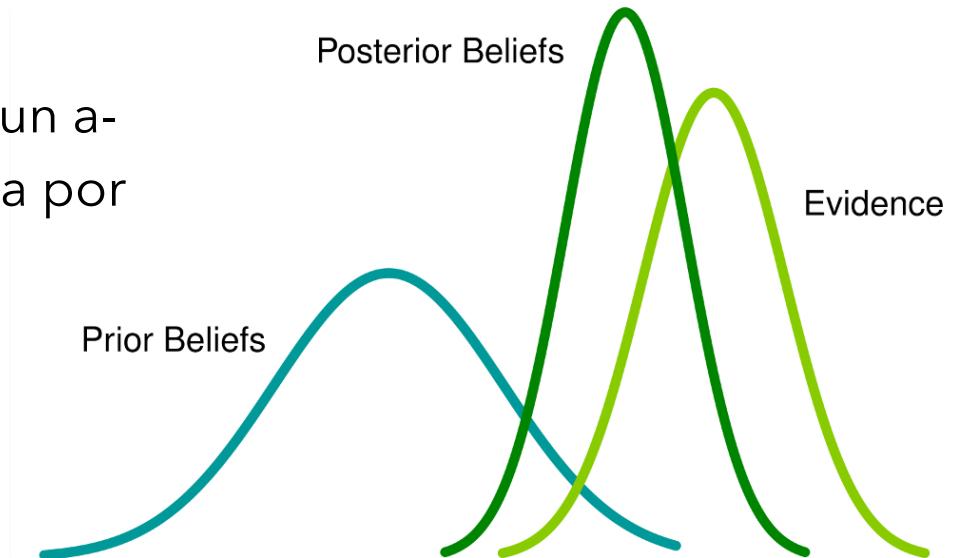
Dropout Variacional

- Extendiendo esta idea, es posible obtener un método donde el a-posteriori (en particular la varianza del ruido multiplicativo α) se puede aprender automáticamente desde el conjunto de entrenamiento usando métodos variacionales.

$$\mathbb{E}_{q_\alpha}(\ell(\theta)) - D_{KL}(q_\alpha(w) || p(w))$$

- El riesgo de overfitting se puede controlar mediante la elección de un a-priori para los pesos, que favorezca por ejemplo distribuciones dispersas.

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$



Fast Dropout

- Uno de los efectos indeseables de Dropout es que tiende a retardar el aprendizaje: se requiere más tiempo y también más iteraciones para obtener convergencia en el conjunto de validación.
- Parte importante del problema es que, en promedio, las neuronas participan de un *backward pass* sólo con probabilidad p , de modo que se pueden requerir muchas más iteraciones para obtener el mismo número de ajustes que se tendrían sin usar Dropout.
- Por ejemplo, si $p = 1/2$, se requieren al menos 5 iteraciones para poder asegurar con probabilidad de al menos 95% que una neurona participa al menos 1 vez del entrenamiento.



Fast Dropout

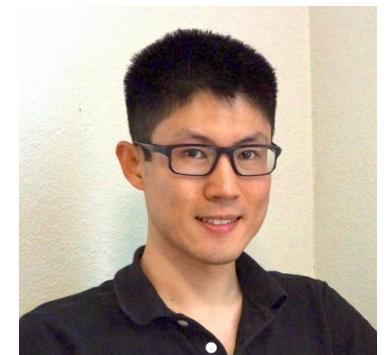
- Wang et al. 2013 proponen resolver este problema aproximando la salida de una neurona entrenada con Dropout durante el entrenamiento. De modo muy similar a la idea que DropConnect aplica después del entrenamiento, tenemos que la pre-activación de una unidad y cuyos inputs x se someten a Dropout, es de la forma

$$y = w^T m \odot x = \sum_j (w_j x_j) m_j$$

Suma pesada de Bernoulli's
 $m_j \sim \text{Ber}(p_i)$

- De nuevo por el TLC (esta vez en versión de Lyapunov) podemos hacer la aproximación

$$y \sim \mathcal{N} \left(\sum_i p_i w_i x_i, \sum_i m_i p_i (1 - p_i) w_i^2 x_i^2 \right)$$



Wang, Sida, and Christopher Manning. "Fast Dropout Training." *International Conference on Machine Learning*. 2013.

Fast Dropout

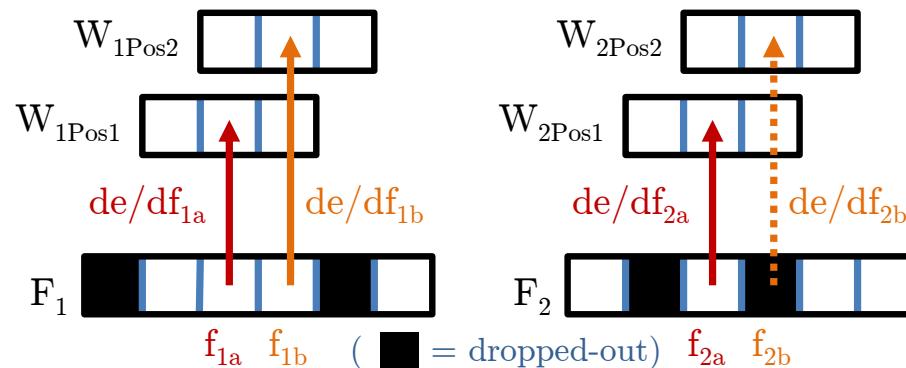
- Muestrear de la Gausiana es aproximadamente equivalente a aplicar múltiples máscaras Dropout. Aún mejor, en vez de muestrear de la Gausiana, los autores muestran que es posible trabajar directamente con su valor esperado y su varianza. Concretamente, para diferentes funciones de activación (logística, ReLu) es posible calcular o aproximar analíticamente el valor esperado de la salida de la neurona si aproximamos la pre-activación mediante la Normal. Por ejemplo, para ReLu $\sigma(z) = \max(0, z)$

$$E_{z \sim \mathcal{N}(\mu, \sigma^2)}[\sigma(z)] = \Phi(\mu/\sigma)\mu + \sigma f_{\mathcal{N}(0,1)}(\mu/\sigma)$$

- Este valor esperado se puede utilizar como sustituto de la salida de una unidad tanto durante el entrenamiento como durante el backward pass, evitando completamente la necesidad de un muestreo.

Dropout para Capas Convolucionales

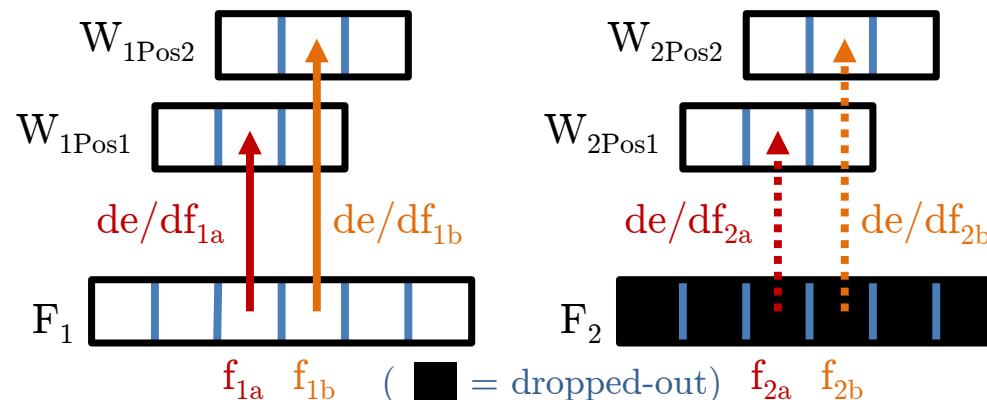
- En el caso de capas convolucionales, aplicar Dropout a cada neurona de un mapa de características (FM) suele no ser muy efectivo.



- Como los “pixeles” de un FM suelen estar muy correlacionados, la información que se oculta enmascarando uno de estos pixeles puede ser transmitida por otros pixeles del FM.

Spatial Dropout

- Como muestra experimentalmente Tompson et al. 2015, en este tipo de capas es mucho más efectivo ocultar un mapa completo de características. Esta variación de Dropout suele denominarse *Spatial Dropout*.



- Normalmente, spatial dropout escala las activaciones por $1/p$ en tiempo de entrenamiento.

Tompson, Jonathan, et al. "Efficient object localization using convolutional networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.



Max-Pooling Dropout

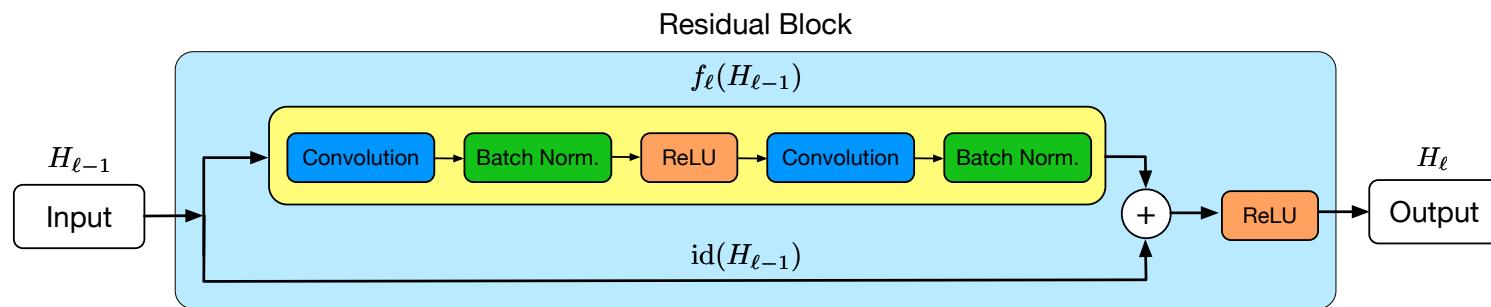
- Wu & Gu proponen en 2015 una capa especial de Pooling que incorpora la idea básica de Dropout. En vez de esconder activaciones o canales entrantes a una capa convolucional, los autores proponen esconder activaciones de un campo receptivo justo antes de aplicar la operación max pooling.
- Una vez entrenada la red, se propone sustituir la operación de pooling por un promedio ponderado de los "pixeles" en el campo receptivo, con pesos iguales a las probabilidades de retención ($q = 1 - p$) de cada pixel.
- Hoy en día es una práctica común (o más "segura") aplicar Dropout sólo después de capas de Pooling, aunque normalmente se acompaña de un escalamiento de las activaciones en vez de una media ponderada.



Wu, Haibing, and Xiaodong Gu. "Towards dropout training for convolutional neural networks." *Neural Networks* 71 (2015): 1-10.

DropLayer

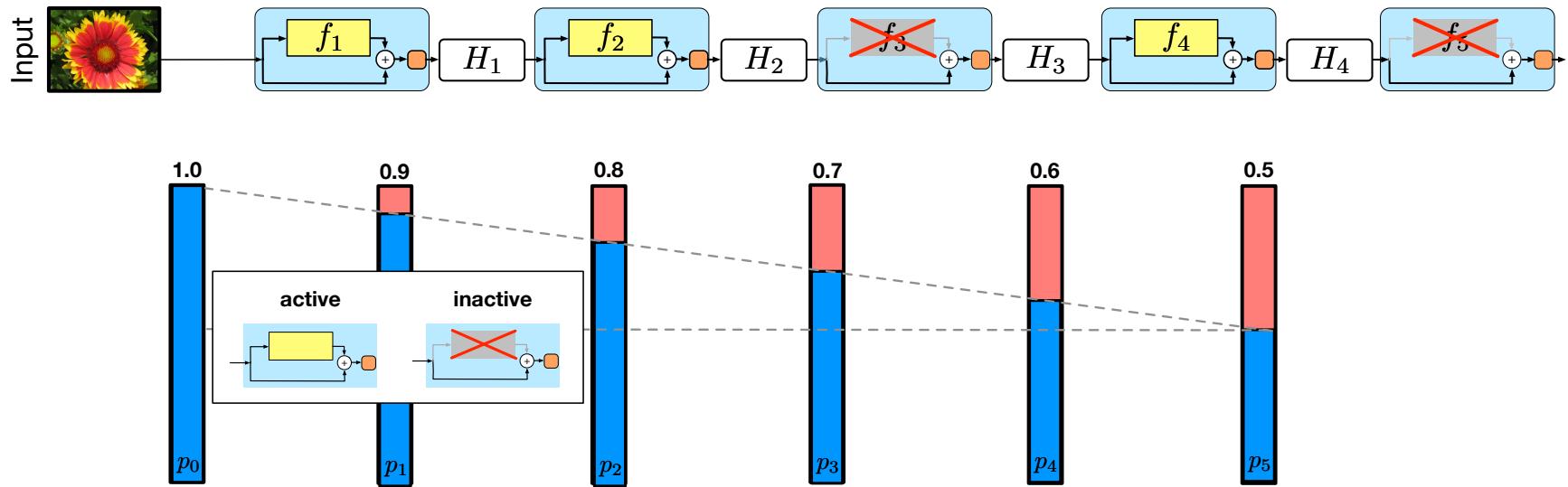
- El reciente desarrollo de redes extremadamente profundas gracias al uso de bloques convolucionales especializados como los que se utilizan en GoogleNet o ResNet ha motivado formas mucho más complejas de Dropout.
- Por ejemplo, Huang et al. proponen en 2016 una forma de Dropout especializada en redes residuales que esconde todo un bloque residual con probabilidad p .



Huang, Gao, et al. "Deep networks with stochastic depth." *European conference on computer vision*. Springer, Cham, 2016.



DropLayer

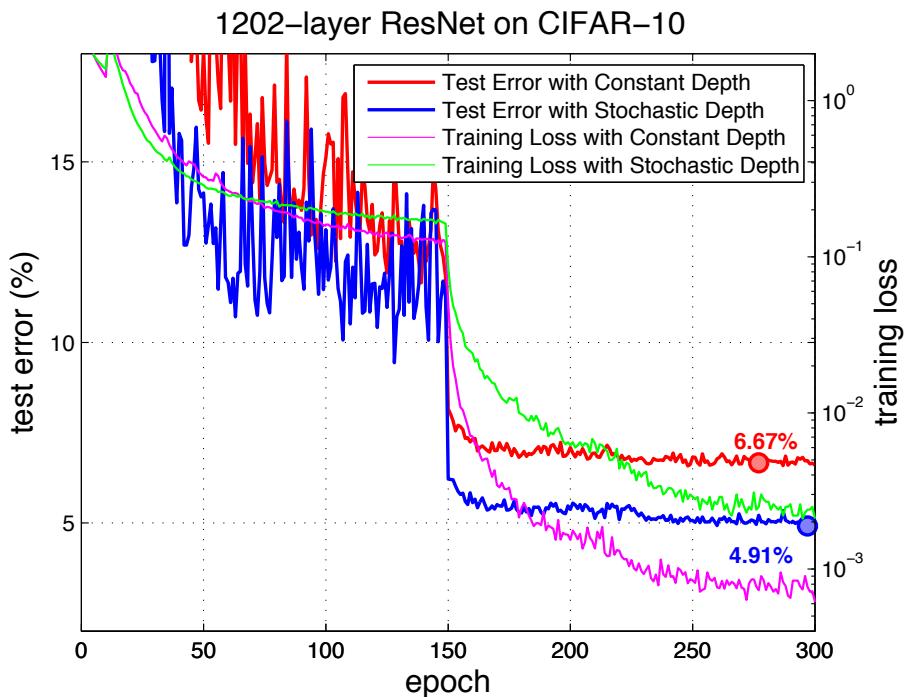
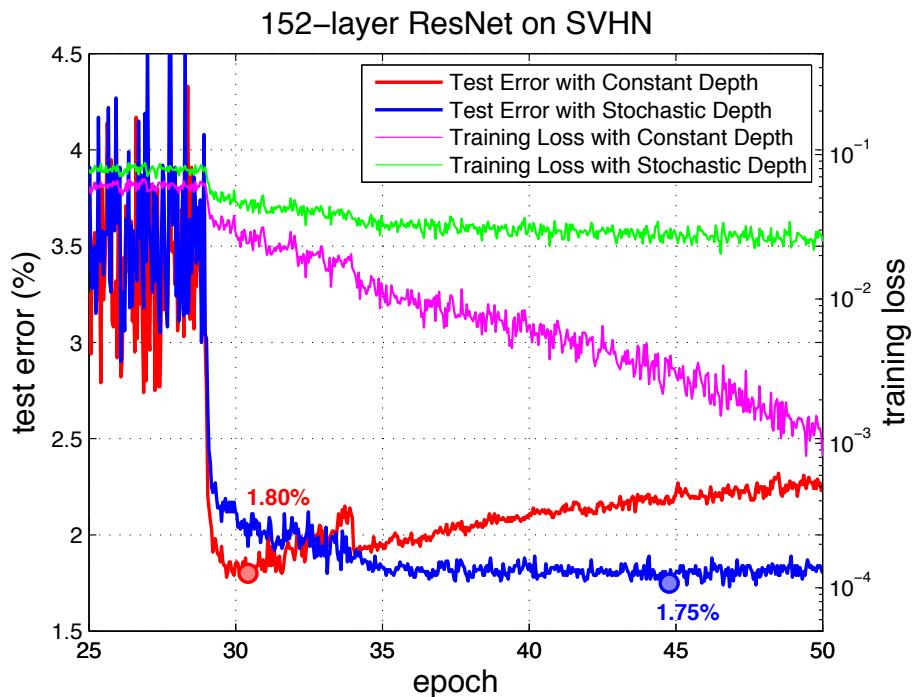


- Los autores proponen utilizar probabilidades de eliminación inversamente proporcionales a la profundidad del bloque y muestran experimentalmente que el método logra regularizar redes extremadamente profundas (100-1000 capas) con gran efectividad.



DropLayer

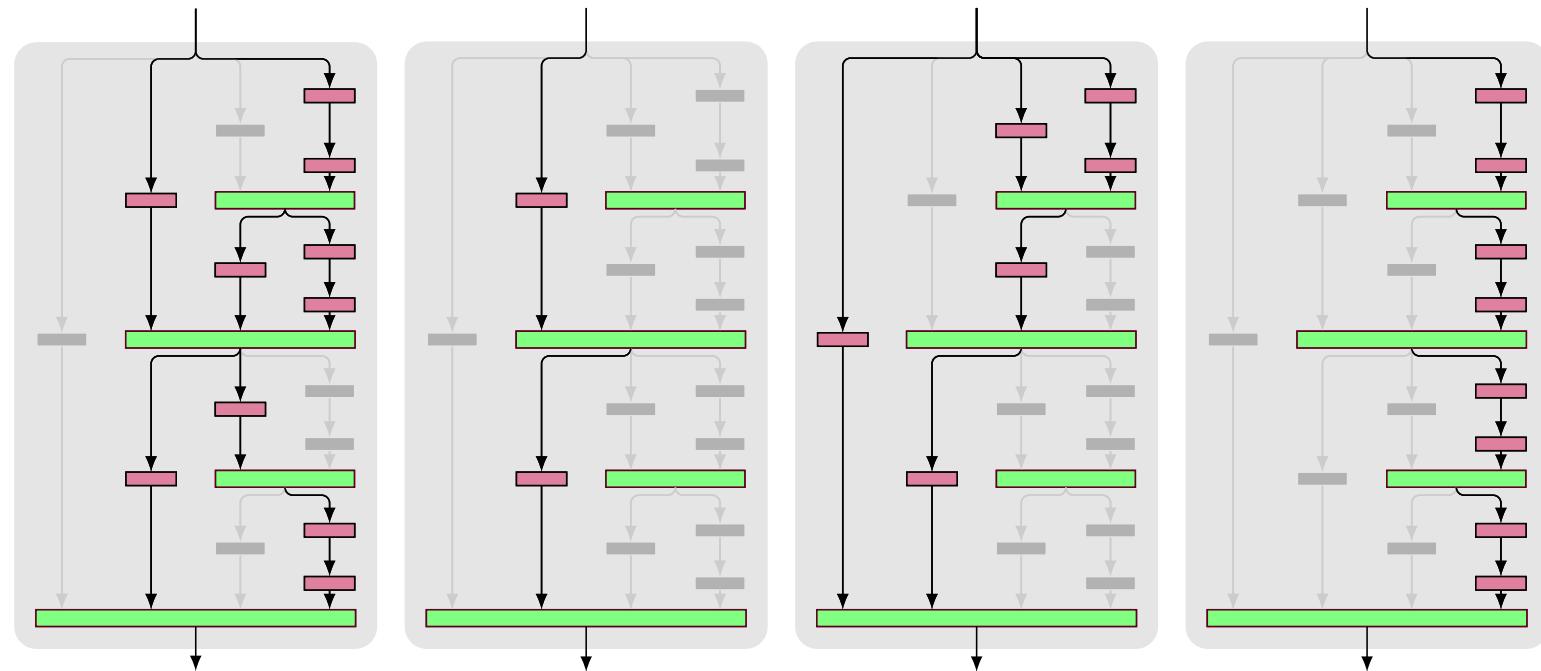
- Uno de los resultados en el trabajo:



- Despues del entrenamiento, la red se opera sin modificaciones.

DropPath

- En 2017, Larsson et al. proponen esconder caminos en la red.



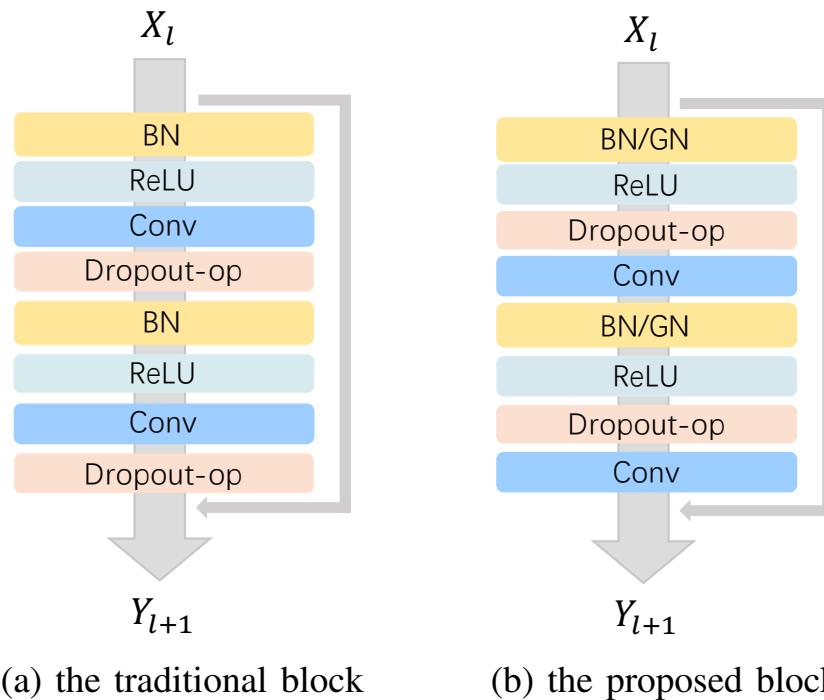
- Experimentos en reconocimiento de imágenes muestran mejoras sobre DropLayer.

Larsson, Gustav, Michael Maire, and Gregory Shakhnarovich.
"Fractalnet: Ultra-deep neural networks without residuals." ICLR 2017.



Patrón Arquitectural Correcto

- Cai et al. han argumentado recientemente que la ineffectividad de Dropout sobre capas convolucionales se debe más que nada al orden en que la operación se combina con otras en redes profundas.



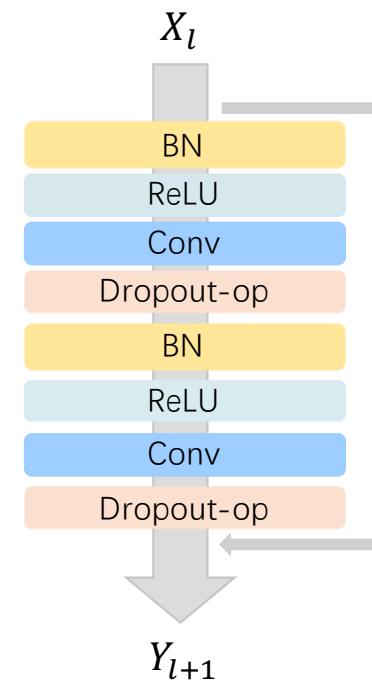
(a) the traditional block

(b) the proposed block

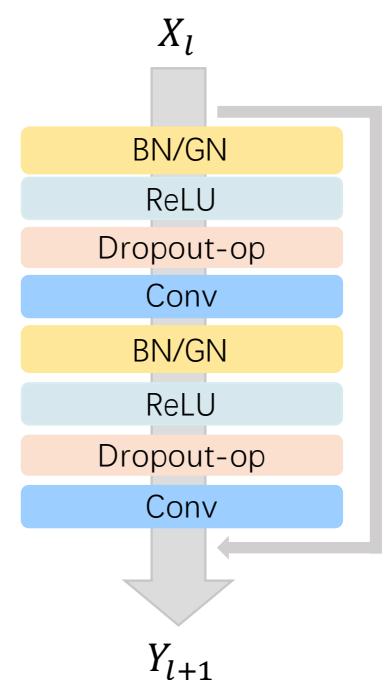
Cai, Shaofeng, et al. "Effective and efficient dropout for deep convolutional neural networks." *arXiv preprint arXiv:1904.03392* (2019).

Patrón Arquitectural Correcto

- En particular, el uso de Dropout antes de una capa de Batch Normalization agrega una varianza significativa a las estadísticas que esta capa calcula en cada iteración de entrenamiento.
- Los autores muestran que el uso de Dropout estándar o Spatial Dropout inmediatamente antes de la capa convolucional es altamente efectivo en la práctica y mucho menos dependiente de la arquitectura que ideas como DropLayer y DropPath.



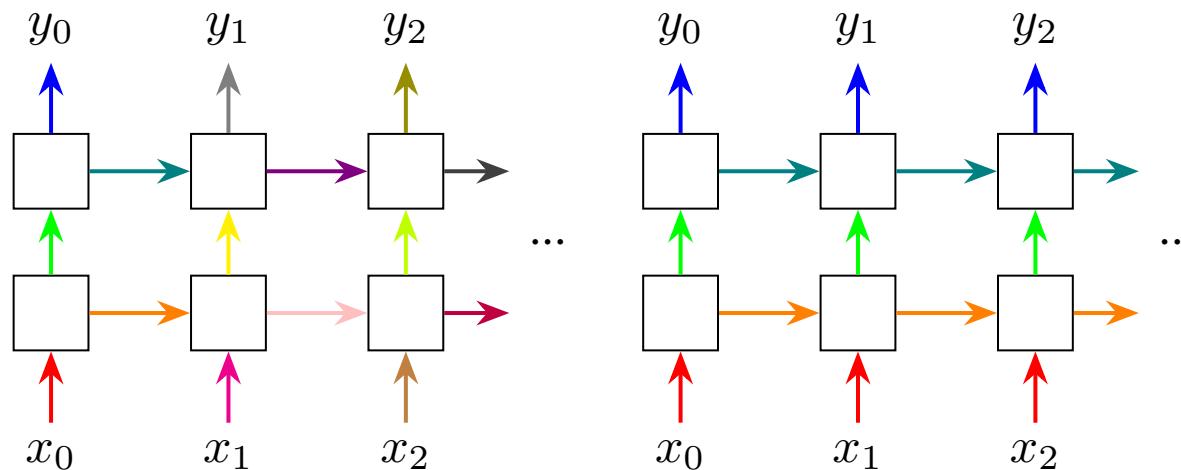
(a) the traditional block



(b) the proposed block

Dropout en Redes Recurrentes

- El uso de Dropout en redes recurrentes requiere también dar atención especial a las características específicas del modelo, pero naturalmente esto lo mencionaremos cuando abordemos ese tema.



Entonces ...

- Dropout es una técnica muy popular hoy en día para regularizar redes neuronales profundas que funciona “apagando” un subconjunto de unidades de la red en cada iteración de entrenamiento.
- Se cree que dropout funcione evitando la co-adaptación de unidades en la red o implementando implícitamente un gran ensemble de sub-redes.
- Las variantes más populares de dropout incluyen: dropconnect y fast dropout. La primera oculta pesos en vez de unidades en cada iteración, mientras que la segunda intenta evitar el costo computacional y el retardo en el entrenamiento que se produce aplicando Dropout explícitamente a la red.
- El uso de dropout de modo “ingenuo” sobre capas convolucionales puede empeorar los resultados en vez de mejorarlo. Se ha propuesto utilizarlo ocultando canales en vez de activaciones puntuales o reservar su uso sólo antes de capas de Pooling.

