

DROPOUT: UMA MANEIRA SIMPLES DE EVITAR OVERFITTING EM REDES NEURAIS

**PALAVRAS-CHAVE: REDES NEURAIS, REGULARIZAÇÃO,
COMBINAÇÃO DE MODELO, DEEP LEARNING**

Nitish Srivastava
Geofrey Hinton
Alex Krizhevsky
Ilya Sutskever
Ruslan Salakhutdinov

Apresentado por: Eduardo Frigini de Jesus

SUMÁRIO

1. Motivação
 2. Inspiração
 3. Dropout
 4. Dropout - Descrição do Modelo
 5. Resultados Experimentais
 6. Weight Decay - Deteriorização do peso
 7. Conclusão
 8. Dropconnect
-

RESUMO

Redes neurais profundas com uma grande quantidade de parâmetros são sistemas de aprendizado de máquinas muito poderosos. **No entanto, o overfitting é um problema sério nessas redes.**

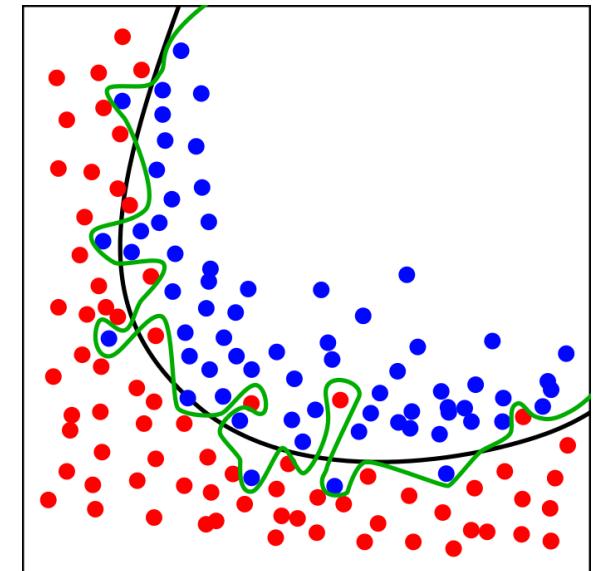
As grandes redes também são lentas de usar, tornando difícil lidar com o overfitting ao combinar as previsões de muitas redes neurais grandes diferentes no momento do teste.

O Dropout é uma técnica para abordar esse problema.

A idéia-chave é retirar aleatoriamente unidades (juntamente com suas conexões) da rede neural durante o treinamento.

Isso evita que as unidades se adaptem demais.

Mostramos que o Dropout melhora o desempenho das redes neurais em tarefas de aprendizagem supervisionadas em visão, reconhecimento de fala, classificação de documentos e biologia computacional.



OVERFITTING

Método dos mínimos quadrados:

$$\sum_{i=1}^n (y_i - \mathbf{z}_i^\top \boldsymbol{\beta})^2$$

Problemas e limitações com uso do método Regressão Linear:

- *Overfitting*
 - Grande quantidade de variáveis
 - Baixa relação de quantidade de exemplos com número de variáveis
- Multicolinearidade
 - Quando uma ou mais variáveis possuem alta correlação
- Sintoma: altos coeficientes

Função Objetiva $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_{p-1} X_{i,p-1} + \varepsilon_i$

OVERFITTING

Técnicas para minimizar os problemas e limitações do método de Regressão Linear:

Regularização

- Penalização no uso dos coeficientes
- Necessário normalizar variáveis

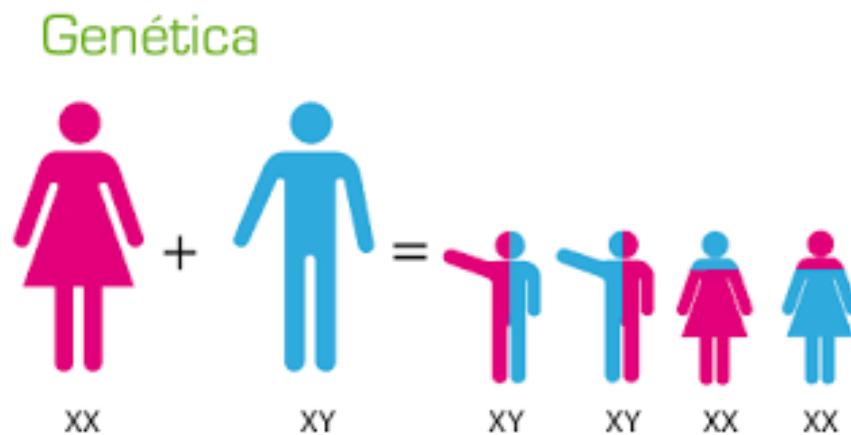
- **Lasso**
$$\sum_{i=1}^n (y_i - \mathbf{z}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p |\beta_j|$$

Funciona como seleção de variáveis

- **Ridge**
$$\sum_{i=1}^n (y_i - \mathbf{z}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p \beta_j^2$$

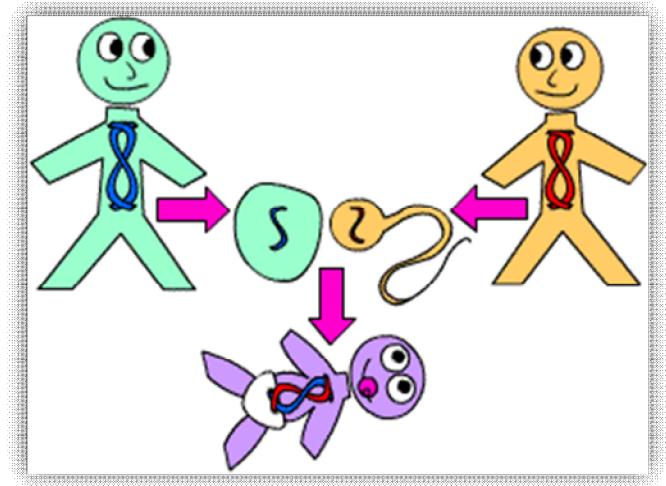
INSPIRAÇÃO

A inspiração para Dropout (Hinton et al., 2013) veio do papel do sexo na evolução.



INSPIRAÇÃO

- Os genes funcionam bem com outro pequeno conjunto aleatório de genes.
- Similarmente, Dropout sugere que cada unidade deve trabalhar com uma amostra aleatória de outras unidades.



DROPOUT

- No treino (cada interação):

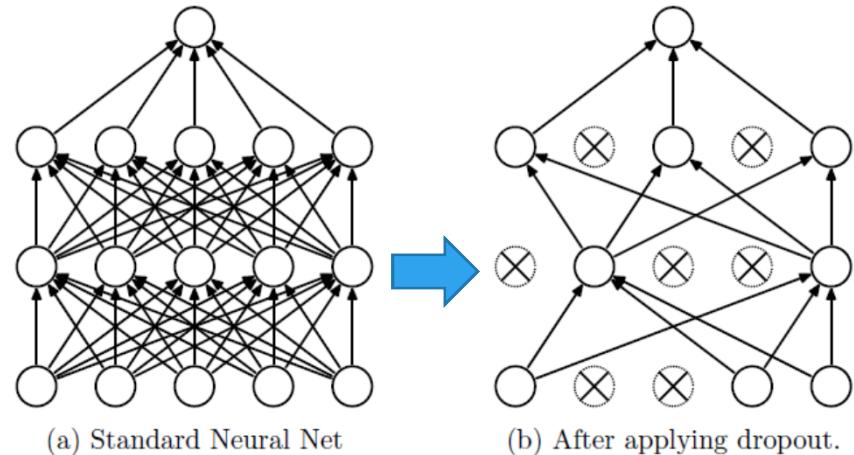
Cada unidade é mantida com uma Probabilidade p .

- No teste:

A rede é usada como um todo.

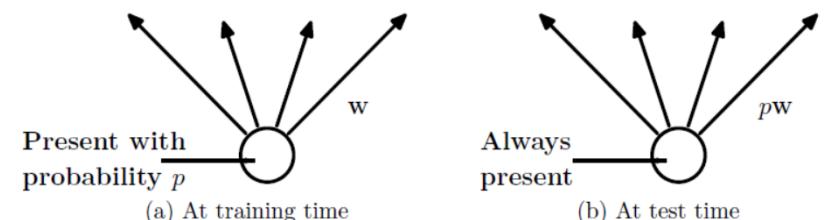
Os pesos são reduzidos por um fator p .

- Na prática, o abandono treina 2^n redes (n – número de unidades).



(a) Standard Neural Net

(b) After applying dropout.



Present with
probability p

(a) At training time

Always
present

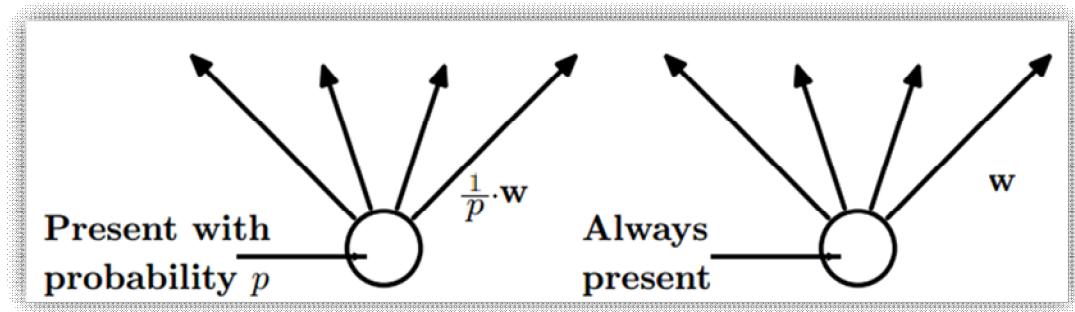
p^W

(b) At test time

$p = 0.5$
UMA BOA OPÇÃO

DROPOUT

- No treino, os pesos são aumentados por um fator de $\frac{1}{p}$
- No momento do teste, nenhuma escala é aplicada.
- Esse método é usado no Tensorflow :
`tf.nn.dropout (x, keep_prob = p)`



DROPOUT – DESCRIÇÃO DO MODELO

A operação feed-forward de uma rede neural padrão é:

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} \mathbf{y}^{(l)} + b_i^{(l+1)}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)})$$

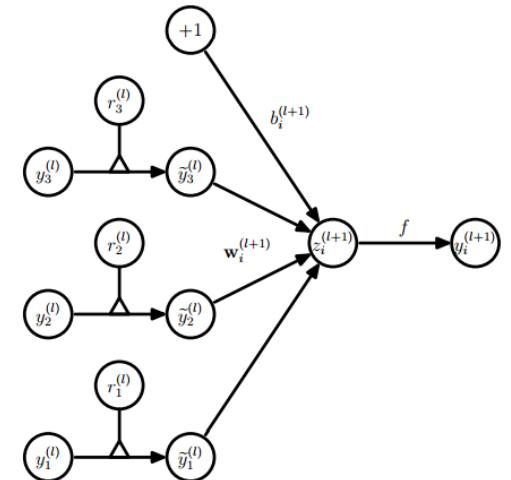
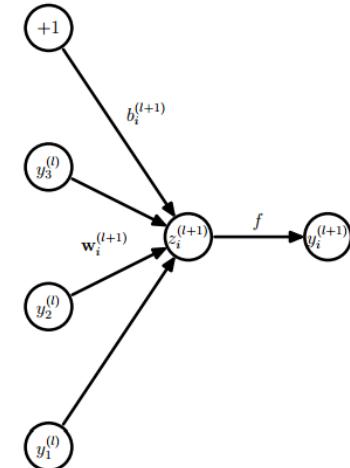
Com dropout, a operação feed-forward torna-se:

$$r_i^{(l)} \sim \text{Bernoulli}(p)$$

$$\tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}$$

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} \tilde{\mathbf{y}}^{(l)} + b_i^{(l+1)}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)})$$



DROPOUT – DESCRIÇÃO DO MODELO

- O backpropagation é executado somente na rede diluída, para cada iteração de treinamento. Um caso de treinamento que não usa um parâmetro, contribui com um gradiente de zero para esse parâmetro.
- No momento de teste, os pesos são dimensionados como:

$$W_{test}^{(l)} = p \cdot W^{(l)}$$

DROPOUT – DESCRIÇÃO DO MODELO

Um olhar mais atento sobre o método da média aproximada:

Vamos suprimir o uso de preconceitos.

Durante o treinamento, nós temos:

$$y^{l+1} = f((M * W) \cdot y^l)$$

Matriz de máscara. Tem colunas de 0's.

Nos testes, nós precisamos computar:

$$y^{l+1} = \sum_M p(M) f((M * W)y^l) \approx f\left(\sum_M p(M)(M * W)y^l\right) = f(pW y^l)$$

Pode ser uma aproximação muito ruim, particularmente para a ativação ReLU.

RESULTADOS EXPERIMENTAIS

Treinamos redes neurais com Dropout para problemas de classificação em conjuntos de dados em diferentes domínios. Descobrimos que o abandono melhorou o desempenho da generalização em todos os conjuntos de dados em comparação com as redes neurais que não utilizaram o Dropout. A Tabela 1 apresenta uma breve descrição dos conjuntos de dados.

- MNIST: um conjunto de dados padrão de dígitos manuscritos.
 - TIMIT: um benchmark de discurso padrão para reconhecimento de voz limpo.
 - CIFAR-10 e CIFAR-100: pequenas imagens naturais (Krizhevsky, 2009).
 - Conjunto de dados de números da casa Street View (SVHN): imagens de números de casas coletados pela Google Street View (Netzer et al., 2011).
 - Reuters-RCV1: uma coleção de artigos da Reuters Newswire.
 - Conjunto de dados alternativos de encapsulamento: recursos de RNA para predição de splicing de genes alternativos (Xiong et al., 2011).
-

RESULTADOS EXPERIMENTAIS

Escolhemos um conjunto diversificado de conjuntos de dados para demonstrar que o Dropout é uma técnica geral para melhorar as redes neurais e não é específico para qualquer domínio de aplicação específico. Nesta seção, apresentamos alguns dos principais resultados que mostram a efetividade do Dropout.

Data Set	Domain	Dimensionality	Training Set	Test Set
MNIST	Vision	784 (28×28 grayscale)	60K	10K
SVHN	Vision	3072 (32×32 color)	600K	26K
CIFAR-10/100	Vision	3072 (32×32 color)	60K	10K
ImageNet (ILSVRC-2012)	Vision	65536 (256×256 color)	1.2M	150K
TIMIT	Speech	2520 (120-dim, 21 frames)	1.1M frames	58K frames
Reuters-RCV1	Text	2000	200K	200K
Alternative Splicing	Genetics	1014	2932	733

Table 1: Overview of the data sets used in this paper.

RESULTADOS EXPERIMENTAIS

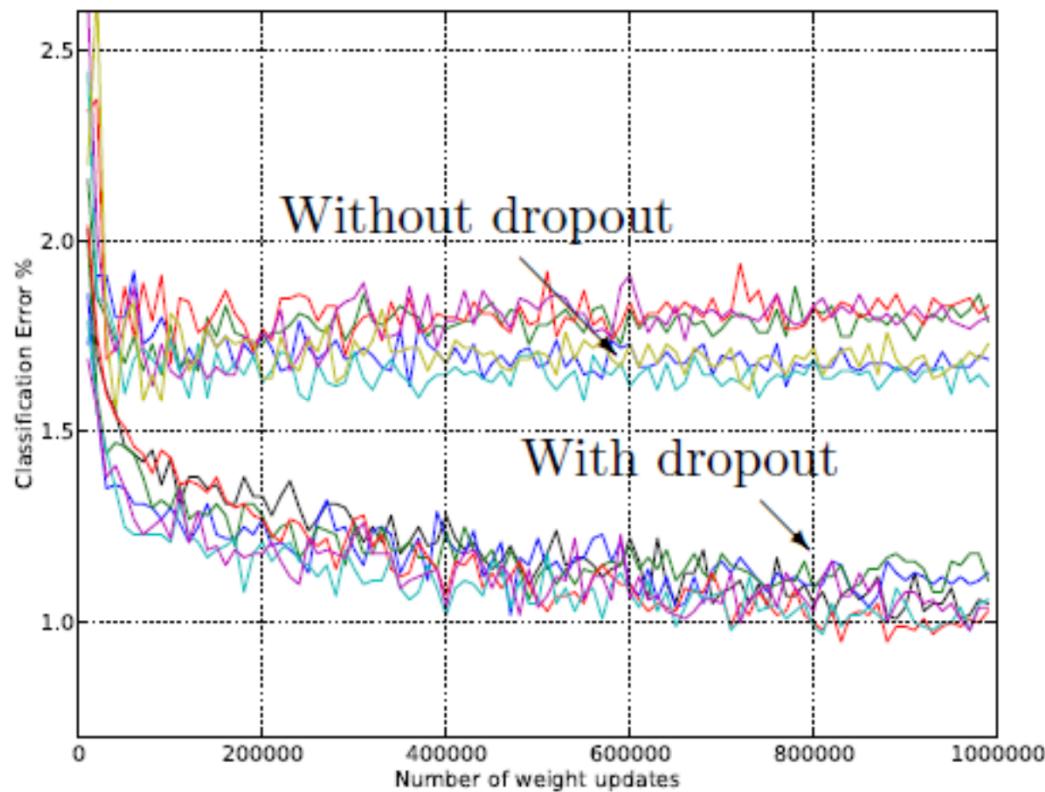
Utilizamos cinco conjuntos de dados de imagem para avaliar o Dropout: MNIST, SVHN, CIFAR-10, CIFAR-100 e ImageNet. Esses conjuntos de dados incluem diferentes tipos de imagens e tamanhos de conjunto de treinamento. Os modelos que obtêm resultados de ponta em todos esses conjuntos de dados usam o Dropout.

Method	Unit Type	Architecture	Error %
Standard Neural Net (Simard et al., 2003)	Logistic	2 layers, 800 units	1.60
SVM Gaussian kernel	NA	NA	1.40
Dropout NN	Logistic	3 layers, 1024 units	1.35
Dropout NN	ReLU	3 layers, 1024 units	1.25
Dropout NN + max-norm constraint	ReLU	3 layers, 1024 units	1.06
Dropout NN + max-norm constraint	ReLU	3 layers, 2048 units	1.04
Dropout NN + max-norm constraint	ReLU	2 layers, 4096 units	1.01
Dropout NN + max-norm constraint	ReLU	2 layers, 8192 units	0.95
Dropout NN + max-norm constraint (Goodfellow et al., 2013)	Maxout	2 layers, (5 × 240) units	0.94
DBN + finetuning (Hinton and Salakhutdinov, 2006)	Logistic	500-500-2000	1.18
DBM + finetuning (Salakhutdinov and Hinton, 2009)	Logistic	500-500-2000	0.96
DBN + dropout finetuning	Logistic	500-500-2000	0.92
DBM + dropout finetuning	Logistic	500-500-2000	0.79

Table 2: Comparison of different models on MNIST.

RESULTADOS EXPERIMENTAIS

Erro de teste para diferentes arquiteturas com e sem Dropout. As redes têm de 2 a 4 camadas ocultas, cada uma com 1024 a 2048 unidades.



RESULTADOS EXPERIMENTAIS

SVHN – Street View House Numbers

- Dropout também é aplicado as camadas convolutivas.
- Todas as camadas ocultas são ReLUs.

Method	Error %
Binary Features (WDCH) (Netzer et al., 2011)	36.7
HOG (Netzer et al., 2011)	15.0
Stacked Sparse Autoencoders (Netzer et al., 2011)	10.3
KMeans (Netzer et al., 2011)	9.4
Multi-stage Conv Net with average pooling (Sermanet et al., 2012)	9.06
Multi-stage Conv Net + L2 pooling (Sermanet et al., 2012)	5.36
Multi-stage Conv Net + L4 pooling + padding (Sermanet et al., 2012)	4.90
Conv Net + max-pooling	3.95
Conv Net + max pooling + dropout in fully connected layers	3.02
Conv Net + stochastic pooling (Zeiler and Fergus, 2013)	2.80
Conv Net + max pooling + dropout in all layers	2.55
Conv Net + maxout (Goodfellow et al., 2013)	2.47
Human Performance	2.0



(a) Street View House Numbers (SVHN)

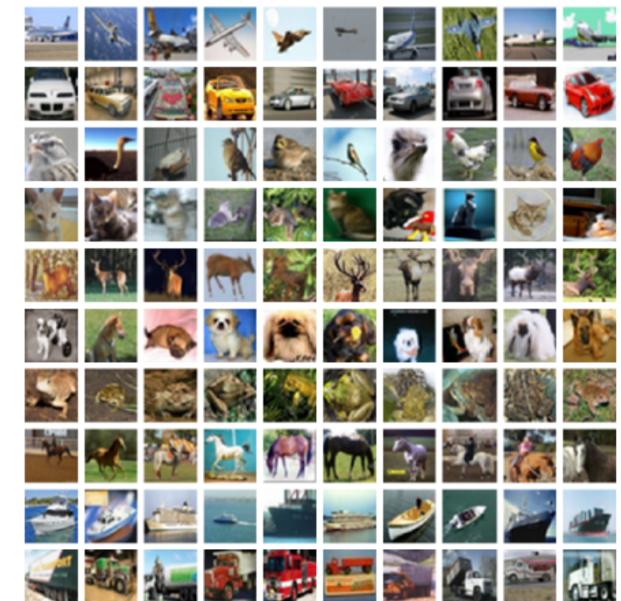
RESULTADOS EXPERIMENTAIS

CIFAR-10 e CIFAR-100:

- Imagens coloridas desenhadas de 32 X 32, de 10 e 100 categorias respectivamente.

Method	CIFAR-10	CIFAR-100
Conv Net + max pooling (hand tuned)	15.60	43.48
Conv Net + stochastic pooling (Zeiler and Fergus, 2013)	15.13	42.51
Conv Net + max pooling (Snoek et al., 2012)	14.98	-
Conv Net + max pooling + dropout fully connected layers	14.32	41.26
Conv Net + max pooling + dropout in all layers	12.61	37.20
Conv Net + maxout (Goodfellow et al., 2013)	11.68	38.57

Table 4: Error rates on CIFAR-10 and CIFAR-100.



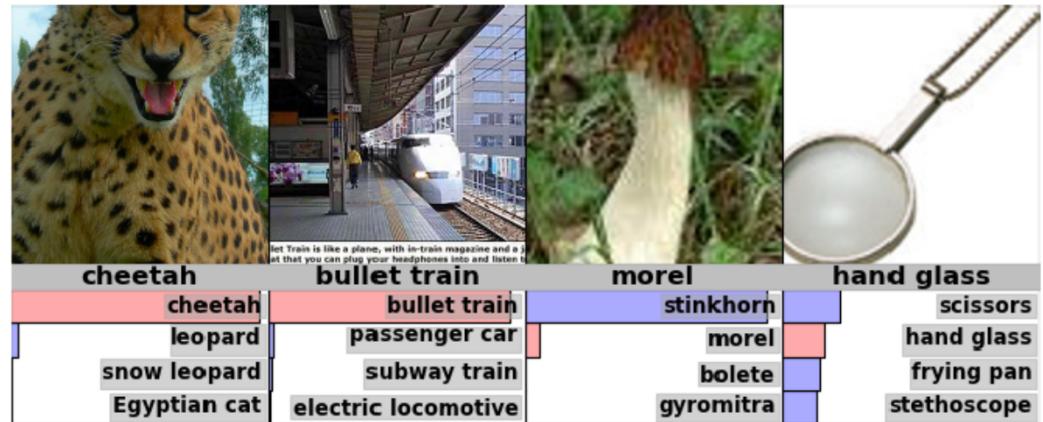
RESULTADOS EXPERIMENTAIS

IMAGINET:

ImageNet: 15MM imagens, alta resolução rotuladas, 22.000 categorias.

ILSVRC (ImageNet Wide-Scale Visual Recognition Challenge): 1000 imagens uma em cada 1000 categorias.

Duas taxas de erro: top-1 e top-5, onde a taxa de erro do top-5 é a fração das imagens de teste para as quais o rótulo correto não está entre os cinco rótulos considerados o mais provável pelo modelo.



RESULTADOS EXPERIMENTAIS

IMAGINET:

ILSVRC-2010 é a única versão do ILSVRC para a qual os rótulos do conjunto de teste estão disponíveis, então a maioria dos nossos experimentos foram realizados nesse conjunto de dados.

As redes convolutivas com Dropout superam outros métodos por uma grande margem.

Model	Top-1	Top-5
Sparse Coding (Lin et al., 2010)	47.1	28.2
SIFT + Fisher Vectors (Sanchez and Perronnin, 2011)	45.7	25.7
Conv Net + dropout (Krizhevsky et al., 2012)	37.5	17.0

Table 5: Results on the ILSVRC-2010 test set.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

Table 6: Results on the ILSVRC-2012 validation/test set.

RESULTADOS EXPERIMENTAIS

TIMIT:

Gravações de 680 alto-falantes, 8 dialetos principais do inglês americano, dez frases foneticamente ricas em um ambiente controlado e sem ruído.

As redes neurais com Dropout foram treinadas em janelas de 21 quadros de log-filter para prever o rótulo da moldura central.

Method	Phone Error Rate%
NN (6 layers) (Mohamed et al., 2010)	23.4
Dropout NN (6 layers)	21.8
DBN-pretrained NN (4 layers)	22.7
DBN-pretrained NN (6 layers) (Mohamed et al., 2010)	22.4
DBN-pretrained NN (8 layers) (Mohamed et al., 2010)	20.7
mcRBM-DBN-pretrained NN (5 layers) (Dahl et al., 2010)	20.5
DBN-pretrained NN (4 layers) + dropout	19.7
DBN-pretrained NN (8 layers) + dropout	19.7

Table 7: Phone error rate on the TIMIT core test set.

RESULTADOS EXPERIMENTAIS

Resultados em um conjunto de dados de texto:

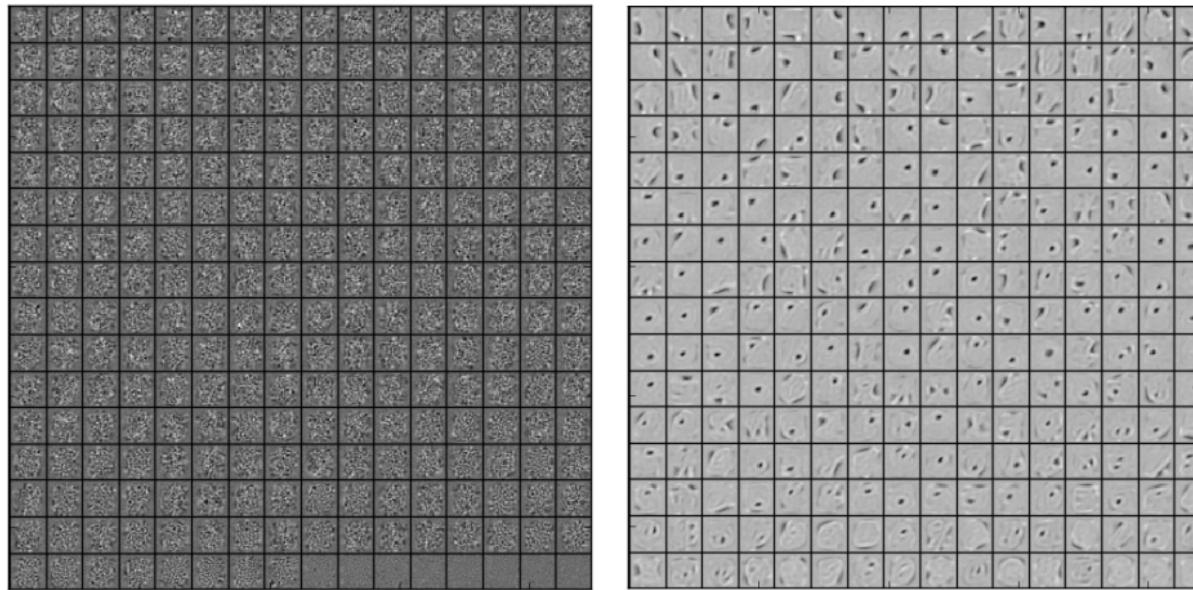
Para testar a utilidade do Dropout no domínio de texto, utilizamos redes com Dropout para treinar um classificador de documentos. Utilizamos um subconjunto do conjunto de dados Reuters-RCV1, que é uma coleção de mais de 800.000 artigos de notícias da Reuters. Estes artigos abrangem uma variedade de tópicos. A tarefa é fazer uma representação de um documento e classificá-lo em 50 tópicos distintos. A nossa melhor rede neural que não utilizou o Dropout obteve uma taxa de erro de 31,05%. A adição de abandono reduziu o erro para 29,62%. Descobrimos que a melhoria foi muito menor em comparação com os conjuntos de dados de visão e fala.

RESULTADOS EXPERIMENTAIS

O efeito do dropout nas características do aprendizado:

- Sem dropout, as unidades tendem a compensar erros de outras unidades.
- Isto leva a overfitting, uma vez que estas co-adaptações não generalizam dados não vistos.
- O Dropout impede co-adaptações, tornando a presença de outras unidades escondidas não confiáveis.

MNIST, uma camada escondida, 256 ReLUs



Sem dropout

As unidades foram adaptadas. Cada unidade não detecta um recurso significativo.

Dropout ($p = 0.5$)

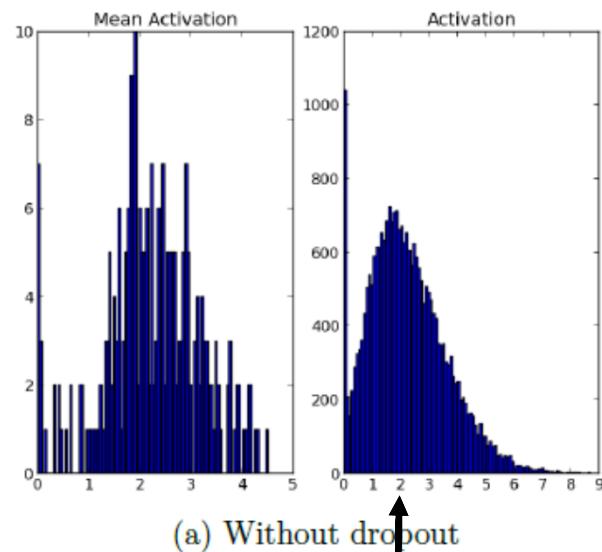
As unidades detectam bordas, traços e manchas em diferentes partes da imagem.

RESULTADOS EXPERIMENTAIS

Efeito na dispersão:

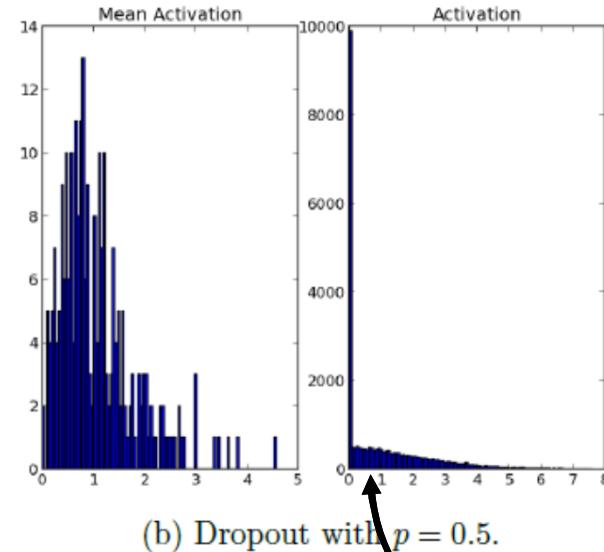
Efeito do Dropout na dispersão da ativação. ReLUs foram usados para ambos os modelos.

O histograma de ativações mostra um modo enorme longe de zero. Uma grande fração de unidades possui alta ativação.



Média de ativação = 2.0

(a) Without dropout



(b) Dropout with $p = 0.5$.

Média de ativação = 0.7

O histograma das ativações mostra um pico acentuado no zero. Poucas unidades têm alta ativação.

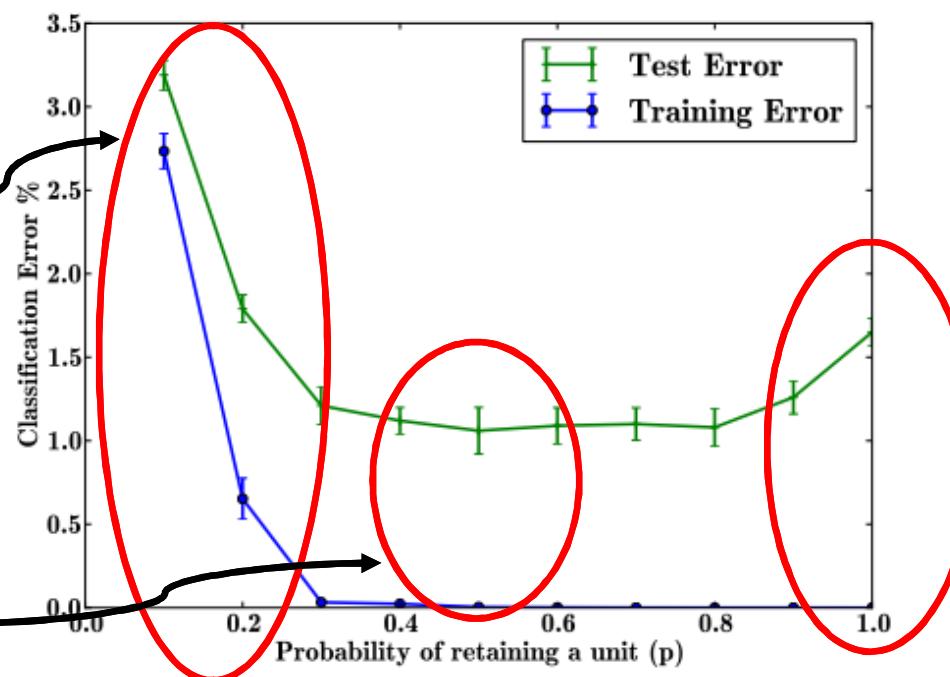
RESULTADOS EXPERIMENTAIS

Os efeitos da taxa p dropout:

- Uma arquitetura 784-2048-2048-2048-10 é usada no conjunto de dados MNIST. A taxa de Dropout p é alterada de números pequenos (a maioria das unidades é descartada) para 1,0 (sem abandono).

Alta taxa de dropout ($p < 0.3$)

- O erro de treino é alto
→ **Underfitting**
- Poucas unidades estão ligadas durante o treino.



Sem dropout ($p = 1.0$)

- O erro de treino é baixo
- O erro de teste é alto
→ **Overfitting**

Melhor taxa de dropout ($p = 0.5$)

- Os erros de treino e teste são baixos
→ **Objetivo alcançado**

RESULTADOS EXPERIMENTAIS

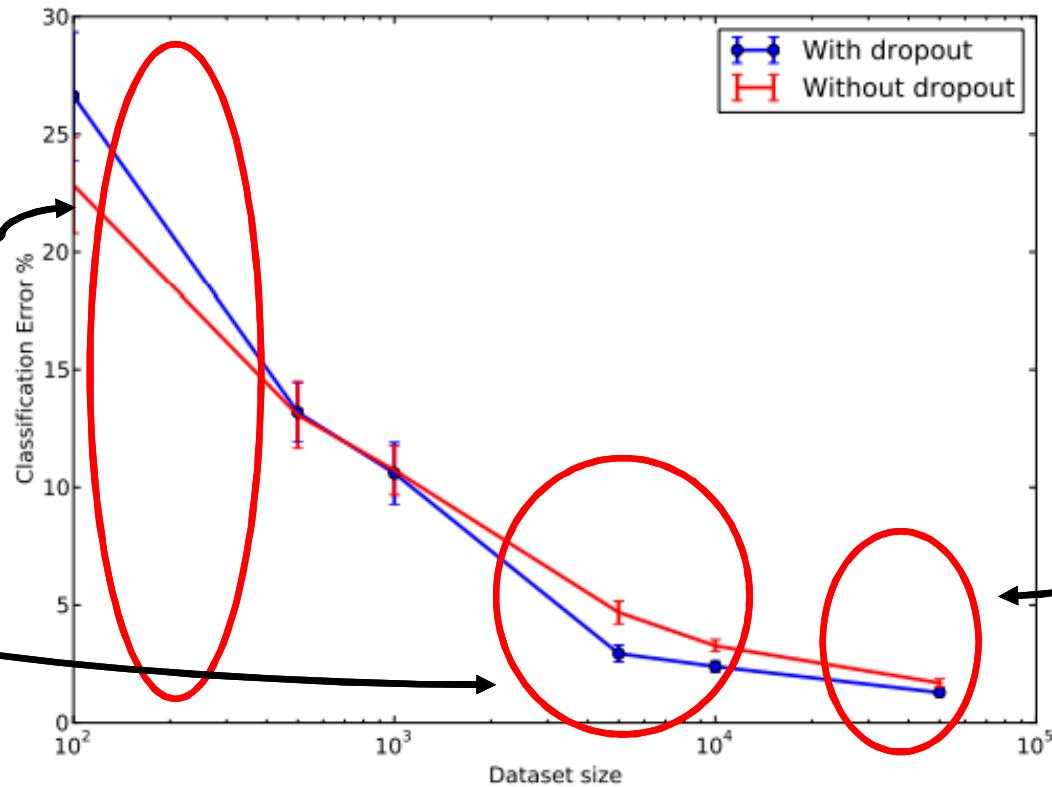
O efeito do tamanho do conjunto de dados

- Uma arquitetura 784-1024-1024-2048-10 é usada no dataset MNIST.

Data set muito pequeno
Dropout não melhora a taxa de erro, e até piora.

Data set médio a grande
Dropout melhora a taxa de erro.

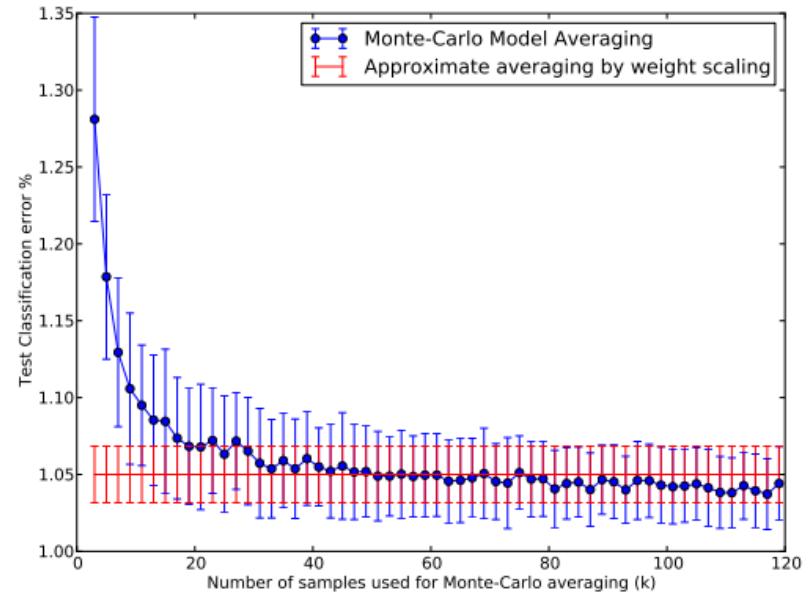
Data set muito grande
Dropout apenas melhora a taxa de erro. O conjunto de dados é suficientemente grande, de modo que a superposição não é um problema.



RESULTADOS EXPERIMENTAIS

Quão boa é a técnica de média aproximada?

- De uma maneira mais correta - experimente muitas redes (k redes) com dropout, e a média de suas saídas ("Monte-Carlo Averaging").
- Usando o conjunto de dados MNIST, em torno de $k = 50$ o método Monte-Carlo torna-se tão bom quanto o método aproximado. Posteriormente, o método de Monte-Carlo é um pouco melhor.



WEIGHT DECAY

- Limitando o crescimento dos pesos na rede.
- Um termo é adicionado à função de perda original, penalizando pesos grandes:

$$\mathcal{L}_{new}(\mathbf{w}) = \mathcal{L}_{old}(\mathbf{w}) + \frac{1}{2}\lambda\|\mathbf{w}\|_2^2$$

- Uma arquitetura de rede 784-1024-1024-2048-10 é usada no conjunto de dados MNIST, com diferentes regularizações.

Method	Test Classification error %
L2	1.62
L2 + L1 applied towards the end of training	1.60
L2 + KL-sparsity	1.55
Max-norm	1.35
Dropout + L2	1.25
Dropout + Max-norm	1.05

WEIGHT DECAY

Dropout tem mais vantagens sobre a deterioração do peso (Helmbold et al., 2016):

- Dropout é livre de escala: dropout não penaliza o uso de grandes pesos quando necessário.
- Dropout é invariante para a escala de parâmetros: dropout não é afetado se os pesos em uma determinada camada forem aumentados por uma constante c , e os pesos em outra camada são reduzidos por uma constante c . Isso implica que o Dropout não possui mínimos locais isolados.

RESULTADOS EXPERIMENTAIS

Comparação com regularizadores padrão

Vários métodos de regularização foram propostos para prevenir o Overfitting em redes neurais.

Estes incluem:

- Weight Decay L2 (mais geralmente, regularização de Tikhonov (Tikhonov, 1943)),
- Lasso (Tibshirani, 1996),
- KL-sparsity e
- regularização máxima-norma.

O Dropout pode ser visto como outra maneira de regularizar as redes neurais. Comparamos o Dropout com alguns desses métodos de regularização usando o conjunto de dados MNIST. A mesma arquitetura de rede (784-1024-1024-2048-10) com ReLUs foi treinada usando descida gradiente estocástica com diferentes regularizações. Os valores de diferentes hiperparâmetros associados a cada tipo de regularização (constantes de decaimento, escassez de alvo, taxa de desistência, limite máximo da norma) foram obtidos usando um conjunto de validação. Descobrimos que o Dropout combinado com a regularização máxima-norma dá o menor erro de generalização.

CONCLUSÃO

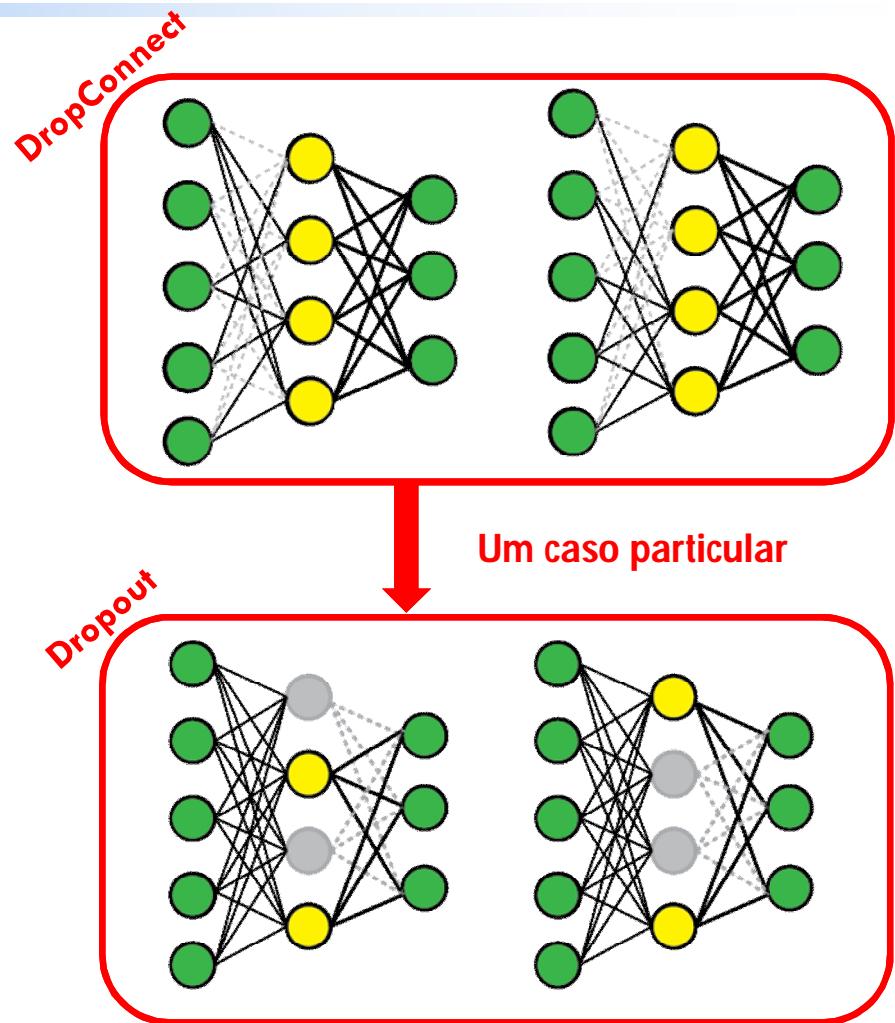
- O Dropout é uma técnica para melhorar as redes neurais reduzindo o overfitting.
- O aprendizado padrão de backpropagation cria co-adaptações frágeis que funcionam para os dados de treinamento, mas não generalizam dados não vistos.
- O Dropout rompe estas co-adaptações, tornando a presença de qualquer unidade escondida particular não confiável.
- Dropout é uma técnica geral e não é específico para nenhum domínio, pode ser usada para: reconhecimento de imagens, fala, documentos, ...
- Os RBMs (Restricted Boltzmann Machine) se encaixam facilmente nesta estrutura.

CONCLUSÃO

- O Dropout é um método de regularização muito bom e rápido.
- O Dropout é um pouco lento para treinar (2-3 vezes mais lento do que sem Dropout).
- É necessário fazer um trade-off entre overfitting e tempo de treinamento.
- Se a quantidade de dados for média-grande o Dropout se destaca. Quando os dados são grandes o suficiente, o Dropout não ajuda muito.
- O Dropout atinge melhores resultados do que os métodos de regularização usados anteriormente (Weight Decay).
- O DropConnect é uma generalização do Dropout. A sua superioridade sobre o Dropout não é clara. Envolve complicações na implementação, e também é mais lento para treinar do que o Dropout.

DROPCONNECT

- DropConnect (Yann LeCun et al., 2013) generaliza o Dropout. Isso sugere descartar pesos.
- Como em Dropout, $p = 0,5$ geralmente dá os melhores resultados.



DROPCONNECT

Técnica de média:

- DropConnect afirma:

$$\begin{aligned} z^{l+1} &= (M * W)y^l \\ E_M[z^{l+1}] &= pW y^l \\ V_M[z^{l+1}] &= p(1 - p)(W * W)(y^l * y^l) \end{aligned}$$

Entrada para a função de ativação

Uma soma ponderada de variáveis de Bernoulli. Pode ser aproximado por uma gaussiana

Estatísticas de gaussiana

- No teste:

- Desenhe amostras de $z^{(l+1)}$ de acordo com a distribuição Gaussiana.
 - Alimente as amostras na função de ativação ($f(z^{(l+1)})$).
 - Calcule a Média.

DROPCONNECT

No-drop, Dropout e DropConnect comparison:

- MNIST: 2 camadas (800 neuronios cada).
- CIFAR-10: 4 camadas. Dropout / DropConnect são aplicados somente no final da camada.
- SVHN: Geralmente algumas arquiteturas como no CIFAR-10. Devido ao grande tamanho do conjunto de treinamento, todos os métodos alcançam a mesma performance.

MNIST

neuron	model	error(%) 5 network	voting error(%)
relu	No-Drop	1.62 ± 0.037	1.40
	Dropout	1.28 ± 0.040	1.20
	DropConnect	1.20 ± 0.034	1.12
sigmoid	No-Drop	1.78 ± 0.037	1.74
	Dropout	1.38 ± 0.039	1.36
	DropConnect	1.55 ± 0.046	1.48
tanh	No-Drop	1.65 ± 0.026	1.49
	Dropout	1.58 ± 0.053	1.55
	DropConnect	1.36 ± 0.054	1.35

CIFAR-10

model	error(%)
No-Drop	23.5
Dropout	19.7
DropConnect	18.7

SVHN

model	error(%) 5 network	voting error(%)
No-Drop	2.26 ± 0.072	1.94
Dropout	2.25 ± 0.034	1.96
DropConnect	2.23 ± 0.039	1.94

DROPCONNECT

No paper DropConnect, eles alcançaram a menor taxa de erro para o MNIST!

crop	rotation scaling	model	error(%) 5 network	voting error(%)
no		No-Drop	0.77±0.051	0.67
		Dropout	0.59±0.039	0.52
		DropConnect	0.63±0.035	0.57
yes	no	No-Drop	0.50±0.098	0.38
		Dropout	0.39±0.039	0.35
		DropConnect	0.39±0.047	0.32
yes	yes	No-Drop	0.30±0.035	0.21
		Dropout	0.28±0.016	0.27
		DropConnect	0.28±0.032	0.21



DROPCONNECT

DropConnect's desvantagens:

- Treinar com DropConnect é mais lento.
- A implementação do DropConnect é mais complicada do que a implementação do Dropout.
- Em seu artigo, DropConnect provou trabalhar principalmente quando se usa mais de uma rede.

A detailed 3D rendering of a neural network. Numerous greyish-blue neurons are interconnected by a complex web of thin, black lines representing synapses. Several of these synapses are highlighted with a bright orange glow, indicating active transmission or firing. The overall effect is one of a dynamic, living brain circuit.

Obrigado