The Implicit and Explicit Regularization Effects of Dropout

Bayesian Inference and Implicit Regularization

Colin Wei [1] Sham Kakade [2,3] Tengyu Ma [1]

Presenters: Kelvin Wong, Siva Manivasagam and Amanjit Singh Kainth

[1]Stanford University [2]Microsoft Research
[3]University of Washington

Dropout[1]

- Regularizes deep networks, especially in vision and language tasks
- Sets a random subset of the activations in each network to 0:

$$\eta = \begin{cases} -1 & \text{with probability } q \\ \frac{q}{1-q} & \text{with probability } 1-q \end{cases}$$

$$h_{\text{drop}} = (\vec{1} + \eta) \odot h$$

[1] Srivastava et al. (2014)

- Why does it work well? Wei et al. (2020) analyze two effects of Dropout.
- Explicit regularization: an explicit change in performance by modifying the objective function ($\mathcal{J}(x)$ vs $\mathcal{J}_{\text{drop}}(x)$)
- Implicit regularization: an implicit change in performance due to stochastic optimization (similar to SGD batch size)
- Let's take a look at how we can disentangle these Dropout effects in training $\rightarrow$ `Google Colab`

Disentangling Explicit and Implicit Regularization

Notation

- Standard objective $\mathcal{J}(x) =: \mathcal{L} \circ F(x)$, decompose $F(x) = F_i(h_i(x))$

- Standard objective $\mathcal{J}(x) =: \mathcal{L} \circ F(x)$, decompose $F(x) = F_i(h_i(x))$
- i-th partial objective $\mathcal{J}_i(h) =: \mathcal{L} \circ F_i(h)$ (note $\mathcal{J}(x) = \mathcal{J}_i(h_i(x))$)

- Standard objective $\mathcal{J}(x) =: \mathcal{L} \circ F(x)$, decompose $F(x) = F_i(h_i(x))$
- i-th partial objective $\mathcal{J}_i(h) =: \mathcal{L} \circ F_i(h)$ (note $\mathcal{J}(x) = \mathcal{J}_i(h_i(x))$)
- Dropout objective (assuming on the i-th hidden layer)

$$
\begin{aligned}
\mathcal{J}_{\text{drop}}(x, \eta) &=: \mathcal{L} \circ F(x, \eta) = \mathcal{L} \circ F_i(h_i(x) + \delta) \\
&= \mathcal{J}_i(h_i(x) + \delta), \quad \delta \triangleq \eta \odot h_i(x) \\
\mathcal{J}_{\text{drop}}(x) &=: \mathbb{E}_\eta\left[\mathcal{J}_{\text{drop}}(x, \eta)\right] \text{ (Expected Dropout objective)}
\end{aligned}
$$

Disentangling Explicit and Implicit Regularization

Notation

- Standard objective $\mathcal{J}(x) =: \mathcal{L} \circ F(x)$, decompose $F(x) = F_i(h_i(x))$
- i-th partial objective $\mathcal{J}_i(h) =: \mathcal{L} \circ F_i(h)$ (note $\mathcal{J}(x) = \mathcal{J}_i(h_i(x))$)
- Dropout objective (assuming on the i-th hidden layer)

$$
\begin{aligned}
\mathcal{J}_{\text{drop}}(x, \eta) &=: \mathcal{L} \circ F(x, \eta) = \mathcal{L} \circ F_i(h_i(x) + \delta) \\
&= \mathcal{J}_i(h_i(x) + \delta), \quad \delta \triangleq \eta \odot h_i(x) \\
\mathcal{J}_{\text{drop}}(x) &=: \mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x, \eta)] \text{ (Expected Dropout objective)}
\end{aligned}
$$

- **Explicit Regularization**: Discrepancy between (expected) training (with Dropout) and standard objectives
- **Implicit Regularization**: Stochasticity-induced, measured by the fluctuation of stochastic gradient dropout around its mean

- Second order approximation to $\mathcal{J}_{\text{drop}}(x, \eta) = \mathcal{J}_i(h_i(x) + \delta)$

$$\mathcal{J}_i(h_i(x) + \delta) \approx \mathcal{J}(x) + \langle \boldsymbol{\nabla} \mathcal{J}_i(h_i(x)), \delta \rangle + \frac{1}{2} \delta^\top \boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)) \delta$$

$$\mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x, \eta)] - \mathcal{J}(x) \approx \frac{1}{2} \left\langle \boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)), \mathbb{E}_\eta\left[\delta \delta^\top\right] \right\rangle$$

Explicit Regularization

Taylor Expansion and Gauss-Newton Approximation

- Second order approximation to $\mathcal{J}_{\text{drop}}(x, \eta) = \mathcal{J}_i(h_i(x) + \delta)$

$$\mathcal{J}_i(h_i(x) + \delta) \approx \mathcal{J}(x) + \langle \boldsymbol{\nabla} \mathcal{J}_i(h_i(x)), \delta \rangle + \frac{1}{2} \delta^\top \boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)) \delta$$

$$\mathbb{E}_\eta [\mathcal{J}_{\text{drop}}(x, \eta)] - \mathcal{J}(x) \approx \frac{1}{2} \left\langle \boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)), \mathbb{E}_\eta \left[\delta \delta^\top\right] \right\rangle$$

- Recall $\delta \triangleq \eta \odot h_i(x)$, can show that $\mathbb{E}_\eta \left[\delta \delta^\top\right] \propto \operatorname{diag}\left(h_i(x)^{\odot 2}\right)$

Explicit Regularization

Taylor Expansion and Gauss-Newton Approximation

- Second order approximation to $\mathcal{J}_{\text{drop}}(x,\eta) = \mathcal{J}_i(h_i(x) + \delta)$

$$\mathcal{J}_i(h_i(x) + \delta) \approx \mathcal{J}(x) + \langle \boldsymbol{\nabla} \mathcal{J}_i(h_i(x)), \delta \rangle + \frac{1}{2} \delta^\top \boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)) \delta$$

$$\mathbb{E}_\eta [\mathcal{J}_{\text{drop}}(x,\eta)] - \mathcal{J}(x) \approx \frac{1}{2} \left\langle \boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)), \mathbb{E}_\eta \left[\delta\delta^\top\right] \right\rangle$$

- Recall $\delta \triangleq \eta \odot h_i(x)$, can show that $\mathbb{E}_\eta \left[\delta\delta^\top\right] \propto \operatorname{diag}\left(h_i(x)^{\odot 2}\right)$
- Use Gauss-Newton approximation on hessian

$$\boldsymbol{\nabla}^2 \mathcal{J}_i(h_i(x)) \approx \boldsymbol{\nabla} F_i(h_i(x))^\top \overbrace{\boldsymbol{\nabla}^2 \mathcal{L}(F(x))}^{H_{\text{out}}(x)} \overbrace{\boldsymbol{\nabla} F_i(h_i(x))}^{J_{F_i}(x)}$$

Explicit Regularization

Taylor Expansion and Gauss-Newton Approximation

- Second order approximation to $\mathcal{J}_{\text{drop}}(x,\eta) = \mathcal{J}_i(h_i(x)+\delta)$

$$\mathcal{J}_i(h_i(x)+\delta) \approx \mathcal{J}(x) + \langle \boldsymbol{\nabla}\mathcal{J}_i(h_i(x)), \delta\rangle + \frac{1}{2}\delta^\top \boldsymbol{\nabla}^2\mathcal{J}_i(h_i(x))\,\delta$$

$$\mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x,\eta)] - \mathcal{J}(x) \approx \frac{1}{2}\left\langle \boldsymbol{\nabla}^2\mathcal{J}_i(h_i(x)), \mathbb{E}_\eta\left[\delta\delta^\top\right]\right\rangle$$

- Recall $\delta \triangleq \eta \odot h_i(x)$, can show that $\mathbb{E}_\eta\left[\delta\delta^\top\right] \propto \operatorname{diag}\left(h_i(x)^{\odot 2}\right)$
- Use Gauss-Newton approximation on hessian

$$\boldsymbol{\nabla}^2\mathcal{J}_i(h_i(x)) \approx \boldsymbol{\nabla}F_i(h_i(x))^\top \overbrace{\boldsymbol{\nabla}^2\mathcal{L}(F(x))}^{H_{\text{out}}(x)}\overbrace{\boldsymbol{\nabla}F_i(h_i(x))}^{J_{F_i}(x)}$$

- Final Form of Explicit Regularizer (multi-layer)

$$R_{\text{approx}}(F,x) \triangleq \sum_i \left\langle J_{F_i}(x)^\top H_{\text{out}}(x)\, J_{F_i}(x), \operatorname{diag}\left(h_i(x)^{\odot 2}\right)\right\rangle$$

Explicit Regularization

Cross-entropy for classification tasks

$$R_{\text{approx}}\left(F,x\right) \triangleq \sum_i \left\langle J_{F_i}\left(x\right)^\top H_{\text{out}}\left(x\right) J_{F_i}\left(x\right), \operatorname{diag}\left(h_i\left(x\right)^{\odot 2}\right)\right\rangle$$

- Full $H_{\text{out}}\left(x\right) \in \mathsf{Mat}\left(|V| \times |V|\right)$, $|V|$ can be quite large!

$$R_{\text{approx}}\left(F,x\right) \triangleq \sum_i \left\langle J_{F_i}\left(x\right)^\top H_{\text{out}}\left(x\right) J_{F_i}\left(x\right), \operatorname{diag}\left(h_i\left(x\right)^{\odot 2}\right)\right\rangle$$

- Full $H_{\text{out}}\left(x\right) \in \mathsf{Mat}\left(|V| \times |V|\right)$, $|V|$ can be quite large!
- For the cross-entropy loss $\mathcal{L}^{\text{ce}}_y\left(\boldsymbol{v}\right) = \mathcal{L}^{\text{ce}}\left(\boldsymbol{v}, y\right) = -\log \mathsf{softmax}\left(\boldsymbol{v}\right)_y$,

$$H^{\text{ce}}_{\text{out}}\left(x\right) = \mathbb{E}_{\widehat{y}\sim\mathsf{softmax}\left(F\left(x\right)\right)}\left[\boldsymbol{\nabla}\mathcal{L}^{\text{ce}}_{\widehat{y}}\left(F\left(x\right)\right)^\top \boldsymbol{\nabla}\mathcal{L}^{\text{ce}}_{\widehat{y}}\left(F\left(x\right)\right)\right]$$

Explicit Regularization

Cross-entropy for classification tasks

$$R_{\text{approx}}\left(F, x\right) \triangleq \sum_i \left\langle J_{F_i}\left(x\right)^\top H_{\text{out}}\left(x\right) J_{F_i}\left(x\right), \operatorname{diag}\left(h_i\left(x\right)^{\odot 2}\right)\right\rangle$$

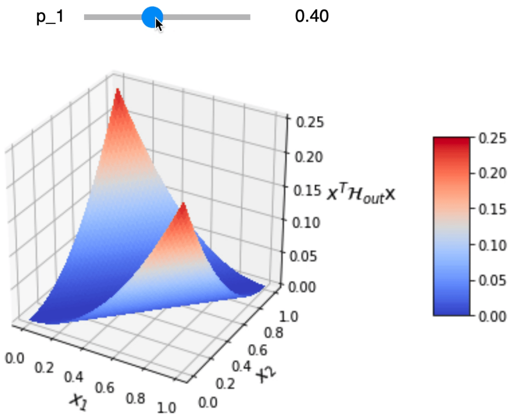
- Full $H_{\text{out}}\left(x\right) \in \mathsf{Mat}\left(|V| \times |V|\right)$, $|V|$ can be quite large!
- For the cross-entropy loss $\mathcal{L}_y^{\text{ce}}\left(\boldsymbol{v}\right) = \mathcal{L}^{\text{ce}}\left(\boldsymbol{v}, y\right) = -\log \mathsf{softmax}\left(\boldsymbol{v}\right)_y$,

$$H_{\text{out}}^{\text{ce}}\left(x\right) = \mathbb{E}_{\widehat{y} \sim \mathsf{softmax}\left(F\left(x\right)\right)}\left[\boldsymbol{\nabla} \mathcal{L}_{\widehat{y}}^{\text{ce}}\left(F\left(x\right)\right)^\top \boldsymbol{\nabla} \mathcal{L}_{\widehat{y}}^{\text{ce}}\left(F\left(x\right)\right)\right]$$

- Memory-efficient (unbiased) estimator for R_{approx}

$$\widehat{R}_{\text{approx}} = \left\langle \boldsymbol{\nabla} \mathcal{J}_i^{\widehat{y}}\left(h_i\left(x\right)\right)^\top \boldsymbol{\nabla} \mathcal{J}_i^{\widehat{y}}\left(h_i\left(x\right)\right), \operatorname{diag}\left(h_i\left(x\right)^{\odot 2}\right)\right\rangle$$

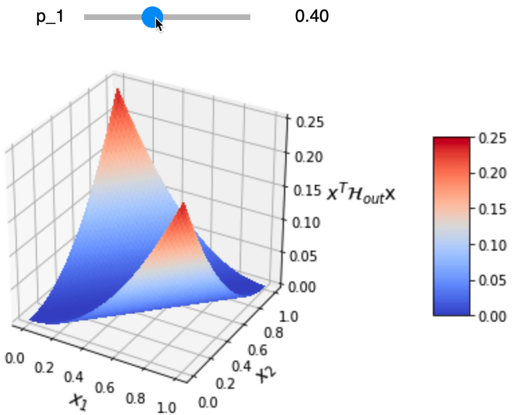
where $\boldsymbol{\nabla} \mathcal{J}_i^{\widehat{y}}\left(h_i\left(x\right)\right) =: \boldsymbol{\nabla} \mathcal{L}_{\widehat{y}}^{\text{ce}}\left(F\left(x\right)\right) J_{F_i}\left(x\right)$ for $\widehat{y} \sim \mathsf{softmax}\left(F\left(x\right)\right)$



$$H^{ce}_{out}(x) = \nabla^2 \mathcal{L}^{ce}_y(F(x)) = \mathrm{diag}(p) - pp^T, \; p = \mathsf{softmax}(F(x))$$

Explicit Regularization

Cross-entropy Hessian



$$H_{\text{out}}^{\text{ce}}(x) = \nabla^2 \mathcal{L}_y^{\text{ce}}(F(x)) = \operatorname{diag}(p) - pp^T, \; p = \mathsf{softmax}(F(x))$$

Intution: Penalize classes with plausible but lower confidence

- Stochastic gradient noise due to Dropout:

$$\xi_{\mathrm{drop}}(F, x, \eta) \triangleq \boldsymbol{\nabla}_W \mathcal{J}_{\mathrm{drop}}(x, \eta) - \boldsymbol{\nabla}_W \mathbb{E}_{\eta'}\left[\mathcal{J}_{\mathrm{drop}}(x, \eta')\right]$$

Implicit Regularization

Identifying the noise in Dropout and approximating it

- Stochastic gradient noise due to Dropout:

$$\xi_{\text{drop}}(F, x, \eta) \triangleq \nabla_W \mathcal{J}_{\text{drop}}(x, \eta) - \nabla_W \mathbb{E}_{\eta'}\left[\mathcal{J}_{\text{drop}}(x, \eta')\right]$$

- Inject noise back by taking the difference in Dropout gradient samples:

$$\widetilde{\xi}_{\text{drop}}(F, x, \eta_1, \eta_2) \triangleq \nabla_W \left[\mathcal{J}_{\text{drop}}(x, \eta_1) - \mathcal{J}_{\text{drop}}(x, \eta_2)\right]$$

Implicit Regularization

Identifying the noise in Dropout and approximating it

- Stochastic gradient noise due to Dropout:

$$\xi_{\text{drop}}(F, x, \eta) \triangleq \boldsymbol{\nabla}_W \mathcal{J}_{\text{drop}}(x, \eta) - \boldsymbol{\nabla}_W \mathbb{E}_{\eta'}\left[\mathcal{J}_{\text{drop}}(x, \eta')\right]$$

- Inject noise back by taking the difference in Dropout gradient samples:

$$\widetilde{\xi}_{\text{drop}}(F, x, \eta_1, \eta_2) \triangleq \boldsymbol{\nabla}_W\left[\mathcal{J}_{\text{drop}}(x, \eta_1) - \mathcal{J}_{\text{drop}}(x, \eta_2)\right]$$

- We saw in the Colab that we can empirically use $\widetilde{\xi}_{\text{drop}}$ to model ξ_{drop}

Implicit Regularization

Taylor Expansion (Linear approximation)

$$\widetilde{\xi}_{\text{drop}}\left(F, x, \eta_i^{(1)}, \eta_i^{(2)}\right) \triangleq \nabla_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(1)}\right) - \nabla_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(2)}\right) \tag{1}$$

- $\mathcal{J}_{\text{drop}}(x, \eta) = \mathcal{J}_i(h_i(x) + \delta)$, where $\delta \triangleq \eta \odot h_i(x)$ is the perturbation on the output of the i-th hidden layer

Implicit Regularization

Taylor Expansion (Linear approximation)

$$\widetilde{\xi}_{\text{drop}}\left(F, x, \eta_i^{(1)}, \eta_i^{(2)}\right) \triangleq \nabla_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(1)}\right) - \nabla_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(2)}\right) \tag{1}$$

- $\mathcal{J}_{\text{drop}}(x, \eta) = \mathcal{J}_i(h_i(x) + \delta)$, where $\delta \triangleq \eta \odot h_i(x)$ is the perturbation on the output of the i-th hidden layer
- First order approximation: $\mathcal{J}_i(h_i(x) + \delta) \approx \overbrace{\mathcal{J}_i(h_i(x))}^{\mathcal{J}(x)} + \langle \nabla \mathcal{J}_i(h_i(x)), \delta \rangle$

Implicit Regularization

Taylor Expansion (Linear approximation)

$$\widetilde{\xi}_{\text{drop}}\left(F, x, \eta_i^{(1)}, \eta_i^{(2)}\right) \triangleq \boldsymbol{\nabla}_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(1)}\right) - \boldsymbol{\nabla}_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(2)}\right) \tag{1}$$

- $\mathcal{J}_{\text{drop}}(x, \eta) = \mathcal{J}_i(h_i(x) + \delta)$, where $\delta \triangleq \eta \odot h_i(x)$ is the perturbation on the output of the i-th hidden layer
- First order approximation: $\mathcal{J}_i(h_i(x) + \delta) \approx \overbrace{\mathcal{J}_i(h_i(x))}^{\mathcal{J}(x)} + \langle \boldsymbol{\nabla} \mathcal{J}_i(h_i(x)), \delta \rangle$
- Thus above simplifies to:

$$\approx \boldsymbol{\nabla}_W \left\langle \boldsymbol{\nabla} \mathcal{J}_i(h_i(x)), (\eta_i^{(1)} - \eta_i^{(2)}) \odot h_i(x) \right\rangle \tag{2}$$

$$\widetilde{\xi}_{\text{drop}}\left(F, x, \eta_i^{(1)}, \eta_i^{(2)}\right) \triangleq \nabla_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(1)}\right) - \nabla_W \mathcal{J}_{\text{drop}}\left(x, \eta_i^{(2)}\right) \tag{1}$$

- $\mathcal{J}_{\text{drop}}(x, \eta) = \mathcal{J}_i(h_i(x) + \delta)$, where $\delta \triangleq \eta \odot h_i(x)$ is the perturbation on the output of the i-th hidden layer
- First order approximation: $\mathcal{J}_i(h_i(x) + \delta) \approx \overbrace{\mathcal{J}_i(h_i(x))}^{\mathcal{J}(x)} + \langle \nabla \mathcal{J}_i(h_i(x)), \delta\rangle$
- Thus above simplifies to:

$$\approx \nabla_W \left\langle \nabla \mathcal{J}_i(h_i(x)), (\eta_i^{(1)} - \eta_i^{(2)}) \odot h_i(x) \right\rangle \tag{2}$$

- Replace $(\eta_i^{(1)} - \eta_i^{(2)})$ with $\sqrt{2}\eta_i$ to maintain same noise covariance

Implicit Regularization

Intuition

$$\xi_{\text{approx}}(F, x, \{\eta_i\}) \triangleq \boldsymbol{\nabla}_W \left(\sum_i \langle \boldsymbol{\nabla} \mathcal{J}_i \left(h_i\left(x\right)\right), \left(\eta_i \odot h_i(x)\right) \rangle \right)$$

- Intuition: provides data-dependent stability during training

- **Explicit** (uses *loss hessian* and *model jacobian*)

$$R_{\text{approx}}\left(F, x\right) \triangleq \sum_i \left\langle J_{F_i}\left(x\right)^\top H_{\text{out}}\left(x\right) J_{F_i}\left(x\right), \text{diag}\left(h_i\left(x\right)^{\odot 2}\right)\right\rangle$$

- **Implicit** (uses *loss jacobian* and *model jacobian*)

$$\xi_{\text{approx}}(F, x, \{\eta_i\}) \triangleq \boldsymbol{\nabla}_W \left(\sum_i \left\langle \boldsymbol{\nabla} \mathcal{J}_i\left(h_i\left(x\right)\right), \left(\eta_i \odot h_i(x)\right)\right\rangle\right)$$

- For the cross-entropy loss $\mathcal{L}^{\text{ce}}$, both penalize the loss and model jacobians $\boldsymbol{\nabla} \mathcal{J}_i\left(h_i\left(x\right)\right) = \boldsymbol{\nabla} \mathcal{L}_y^{\text{ce}}\left(F\left(x\right)\right) J_{F_i}\left(x\right)$ (in practice)

Colab

Let's take a look at these regularizers in action

Conclusion

Some key takeaways

- Dropout induces distinct regularization effects:

Conclusion

Some key takeaways

- Dropout induces distinct regularization effects:
    - **Explicit Regularization:** Induced by change in training objective.

$$\mathbb{E}_x[\mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x, \eta)] - \mathcal{J}(x)] \tag{3}$$

Conclusion

Some key takeaways

- Dropout induces distinct regularization effects:
  - **Explicit Regularization:** Induced by change in training objective.

$$\mathbb{E}_x[\mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x,\eta)] - \mathcal{J}(x)] \tag{3}$$

  - **Implicit Regularization:** Induced by stochastic approximation of Dropout training objective.

$$\nabla_W \mathcal{J}_{\text{drop}}(x,\eta) - \nabla_W \mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x,\eta)] \tag{4}$$

Conclusion

Some key takeaways

- Dropout induces distinct regularization effects:
  - **Explicit Regularization:** Induced by change in training objective.

$$\mathbb{E}_x[\mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x,\eta)] - \mathcal{J}(x)] \tag{3}$$

  - **Implicit Regularization:** Induced by stochastic approximation of Dropout training objective.

$$\nabla_W \mathcal{J}_{\text{drop}}(x,\eta) - \nabla_W \mathbb{E}_\eta[\mathcal{J}_{\text{drop}}(x,\eta)] \tag{4}$$

- We can derive analytical forms of these regularizers, revealing:
  - Explicit regularization encourages output Jacobians and hidden layers to be small according to output Hessian.
  - Implicit regularization penalizes loss Jacobian and hidden layers.

Conclusion

Limitations and outlook

- Interpretation of Dropout's implicit regularizer remains opaque.

Conclusion

Limitations and outlook

- Interpretation of Dropout's implicit regularizer remains opaque.
  - It is unclear why gradient noise helps generalization.

Conclusion

Limitations and outlook

- Interpretation of Dropout's implicit regularizer remains opaque.
  - It is unclear why gradient noise helps generalization.
  - On larger datasets and/or other model architectures, the benefits of Dropout's implicit regularization effect is missing.

Conclusion

Limitations and outlook

- Interpretation of Dropout's implicit regularizer remains opaque.
  - It is unclear why gradient noise helps generalization.
  - On larger datasets and/or other model architectures, the benefits of Dropout's implicit regularization effect is missing.
  - In fact, implicit regularization sometimes makes things worse.

- Interpretation of Dropout's implicit regularizer remains opaque.
  - It is unclear why gradient noise helps generalization.
  - On larger datasets and/or other model architectures, the benefits of Dropout's implicit regularization effect is missing.
  - In fact, implicit regularization sometimes makes things worse.
- The derived analytical regularizers are totally impractical.

Conclusion

Limitations and outlook

- Interpretation of Dropout's implicit regularizer remains opaque.
  - It is unclear why gradient noise helps generalization.
  - On larger datasets and/or other model architectures, the benefits of Dropout's implicit regularization effect is missing.
  - In fact, implicit regularization sometimes makes things worse.
- The derived analytical regularizers are totally impractical.
- But this work provides intuition on Dropout's inner workings, hopefully leading to better, more principled regularizers.

Conclusion

Thanks for listening! Questions?

References I

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15 (56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Colin Wei, Sham Kakade, and Tengyu Ma. The implicit and explicit regularization effects of dropout. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10181–10192. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/wei20d.html.

Appendix: Explicit Regularization

Additional notes: Using the loss hessian is important

$$R_{\text{approx}}(F, x) \triangleq \sum_i \left\langle J_{F_i}(x)^\top \overbrace{H_{\text{out}}}^{\text{substitute}}(x)\, J_{F_i}(x), \operatorname{diag}\left(h_i(x)^{\odot 2}\right) \right\rangle$$

Training Method	Best Val. Ppl.
ℓ_2 reg (tuned)	112.04
$\mathbf{I}$	108.76
$\widetilde{R}_{\text{approx}}$ (tuned)	84.06
R_{approx}	84.52

Table: Effect of explicit regularizer only

- Using identity $\mathbf{I}$ (classical GN matrix) removes most of Dropout benefits
- Using loss Jacobian: $\widetilde{R}_{\text{approx}}(F, x) \triangleq \sum_i \boldsymbol{\nabla}\mathcal{J}_i(h_i(x))\operatorname{diag}(h_i(x)^{\odot 2})\boldsymbol{\nabla}\mathcal{J}_i(h_i(x))^\top$
  $\rightarrow$ *differs from R_{approx} implementation by using label y instead of sample $\widehat{y}$*

Additional notes: Dataset dependent

For large datasets, k-Dropout performs the same or better than the original Dropout.

Table: Experimental results on the full WikiText-103 dataset for QRNN architecture.

Training Method	Best Val. Ppl.
Dropout_1	34.24
Dropout_2	33.35
Dropout_4	32.74
Dropout_8	32.78

More study required to better understand implicit regularization - different effects might be inter-connected.