Introduction to Knowledge Distillation

Smaller, Smarter AI Models for the edge

David Selinger

CEO & Founder

Deep Sentinel

Deep Sentinel

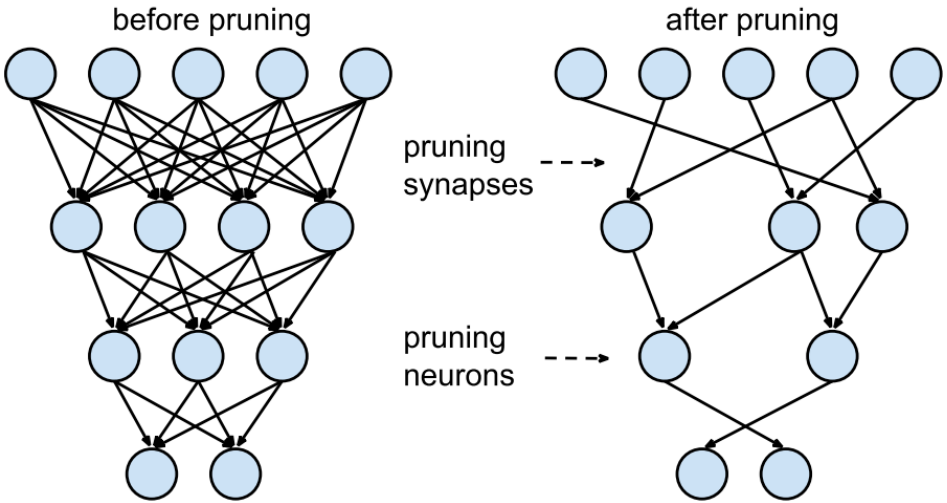
Making smaller, faster models.

v1.0: Pruning

Deep Sentinel

© Copyright Deep Sentinel

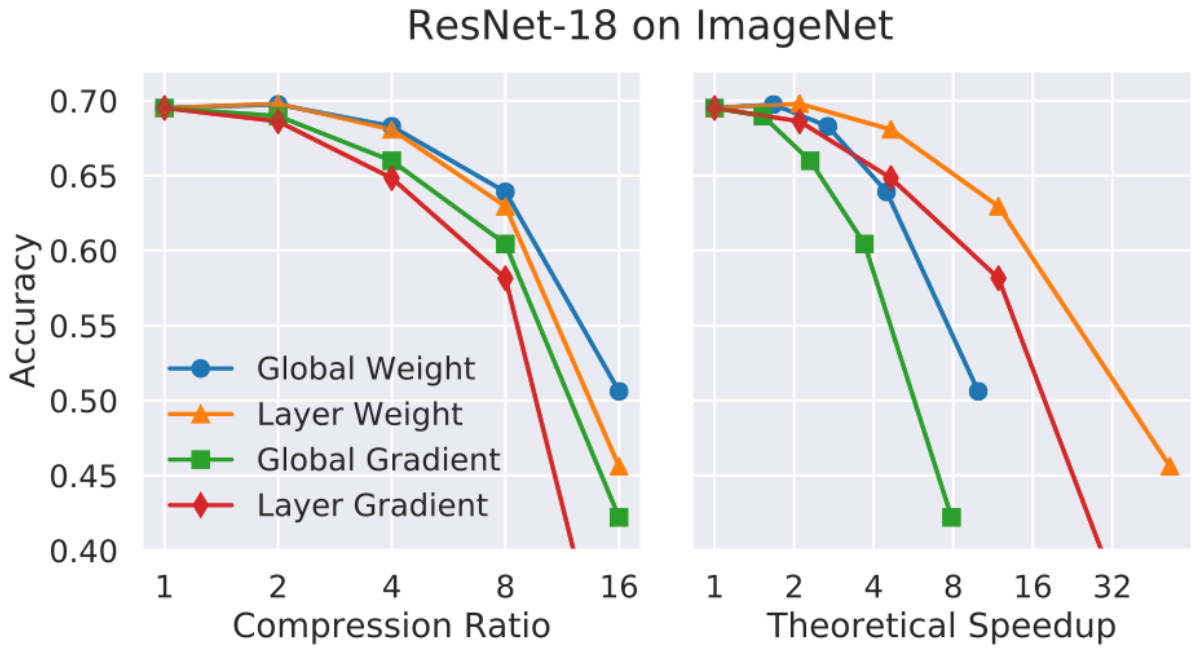
Neural Network Density: Pruning



Deep Sentinel

© Copyright Deep Sentinel

Neural Network Density: Pruning



Insight: ***Not all neurons are equally valuable for knowledge-gain,***
https://blog.paperspace.com/neural-network-pruning-explained/

Deep Sentinel

© Copyright Deep Sentinel

Making smaller, faster models.

v2.0: Distillation

Deep Sentinel

© Copyright Deep Sentinel

What is Knowledge Distillation

- A form of “transfer learning”
  - From one *architecture* to another
  - From one *network* size to another
  - From one *initialization* state to another
  - From one *domain* to another

- In practice: Model compression technique
  - “Knowable” loss of precision
  - Target memory, compute, power budgets
  - Leverages pre-trained models as a *starting point*
  - Can target hardware/software-*optimized architectures*

© Copyright Deep Sentinel

Distillation: Why is This Important?

- **HARDWARE:** The embedded lifestyle: Compute, power, storage, memory limitations
  - Current bottleneck: Memory
- **SOFTWARE:** Lots of model-types/architectures: Numerous chipsets with different performance characteristics. How do I build something for *my* solution/device?
- **HUMAN RESOURCES:** Distillation does not require a PhD or 10+ years of experience

© Copyright Deep Sentinel

PARTICIPANT TIME!

WHO LIKES FROGS?

© Copyright Deep Sentinel

FROGS OF THE AMAZON RAINFOREST

Deep Sentinel

© Copyright Deep Sentinel

The Poison Dart Frog

- Yellow and/or orange back striped with black
- Light Blue mottled belly
- Blue appendages may be mottled also

Deep Sentinel

© Copyright Deep Sentinel

Is this a Poison Dart Frog?

© Copyright Deep Sentinel

Am I Poisoned Dart frog?

Deep Sentinel

© Copyright Deep Sentinel

Am I?

© Copyright Deep Sentinel

ME?

Deep Sentinel

© Copyright Deep Sentinel

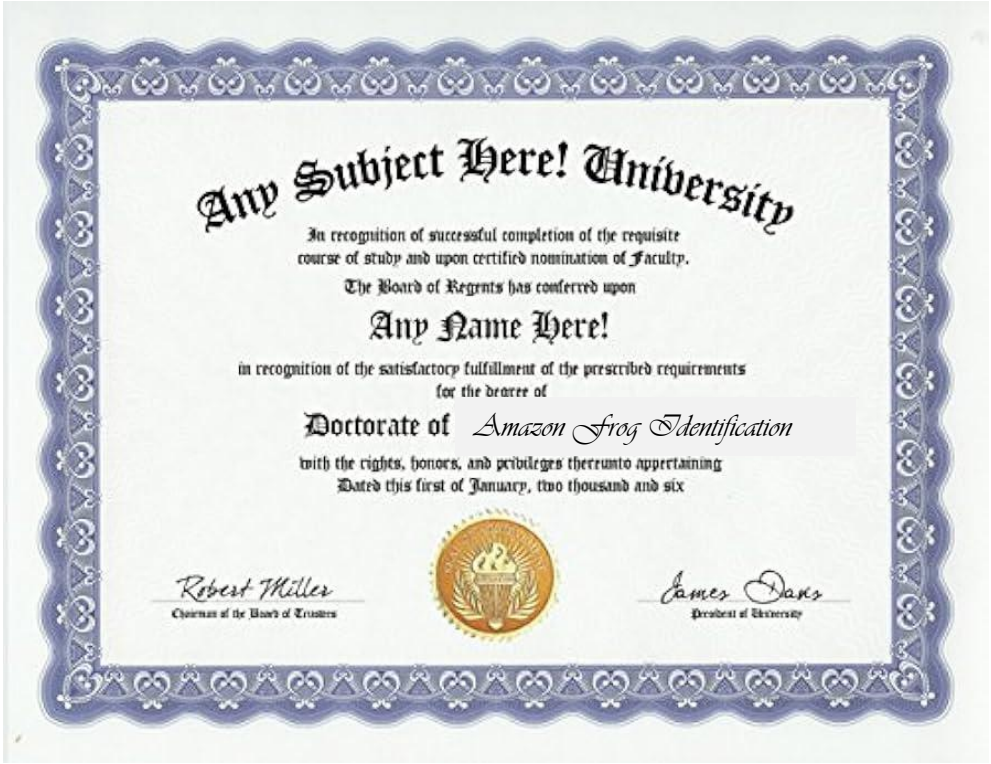
How about this one?

© Copyright Deep Sentinel

OK FINAL TEST!?!

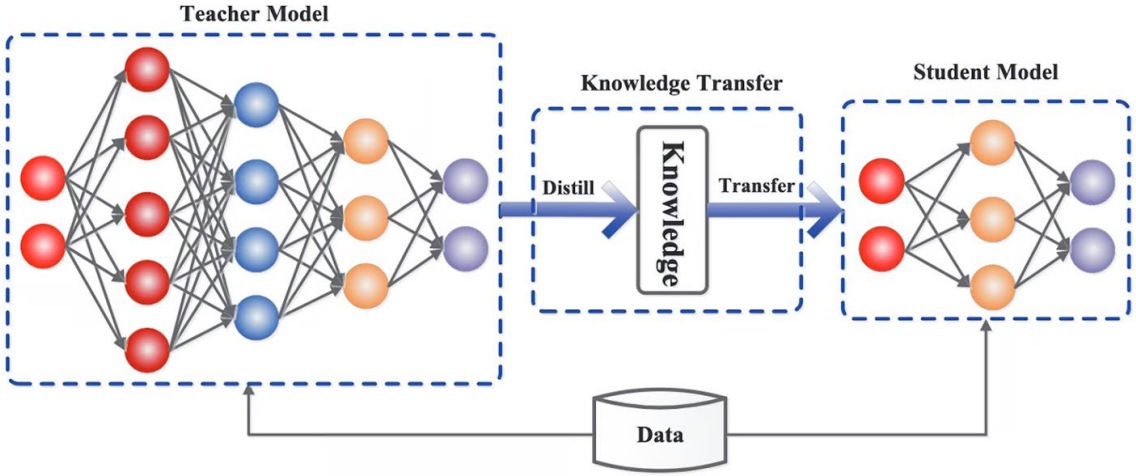
© Copyright Deep Sentinel

Congratulations!!!



© Copyright Deep Sentinel

How Does It Work?



Lots of features
Multiple/different domains
Long compute
Origin architecture (pre-trained)

Fewer (better) Features
Domain-specific
Fewer steps
Any architecture

Ref: https://www.datacamp.com/blog/distillation-llm

© Copyright Deep Sentinel

My first Experience with Distillation: ImageNet as Teacher Applied to Security (2016)

EMBEDDED VISION SUMMIT

Teacher

- DataSet: ImageNet-22K
- Model architecture: Resnet, Inception (SOTA 2015)

(Very) Slow, but ~80% Accuracy

Student

- DataSet: 66k CCTV images with “Security Objects” tagged (Cars, people, animals, doors)
- Hand-tagged (took 1 week)
- Model architecture: Inception (hardware optimized)

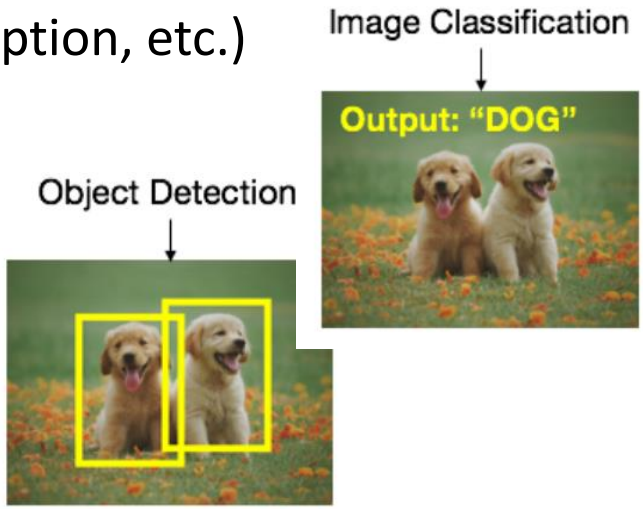
Result: 75% accuracy after 10 epochs

Deep Sentinel

© Copyright Deep Sentinel

ML Applications of Distillation

- Object classification: ImageNet (ResNet, Inception, etc.)
- Object detection: Yolo, MobileNet, DarkNet
- LLM Distillation: LLama, DeepSeek
- Diffusion models



Next-up: How can we implement distillation?

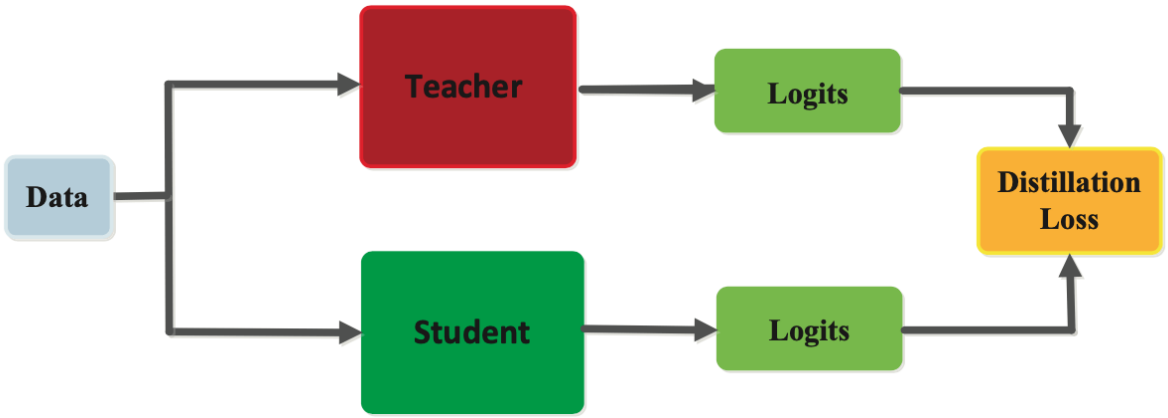
- *Critical Insight:* Sometimes students can outperform the teachers
- We'll use a classification example to explore how to distill network knowledge

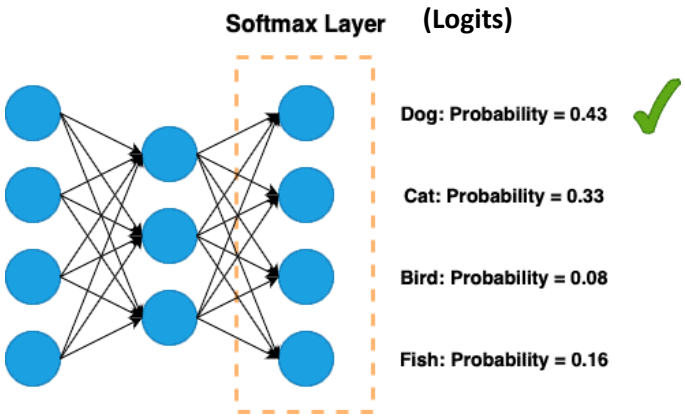
https://www.oreilly.com/library/view/neural-network-projects/9781789138900/e1f93bb9-0e51-428d-8e06-f19143ecc927.xhtml

Deep Sentinel

© Copyright Deep Sentinel

Training: A Basic Distillation





- Logits as Target: The final prediction layer
- Loss functions:
  - Classification: Cross Entropy (standard classification loss) +
  - Distillation: KL divergence (distillation difference)

https://neptune.ai/blog/knowledge-distillation
https://www.baeldung.com/cs/softmax-vs-log-softmax

Deep Sentinel

© Copyright Deep Sentinel

Training Pseudocode

- For (*epochs*) do:
  - for each (x, y) ε D
    - $y_t = T(x)$
    - $y_s = S(x)$
    - Loss = Cross-Entropy Loss (y, y_s) + KL Divergence (y_t, y_s)
    - BackProp (S, Loss)

D = Dataset of target domain
T = Teacher network (forward compute to logits)
S = Student network

Insights:

How does D affect the training?

- What if D is a subset of T's domain?
- What if D has no overlap of T's domain?

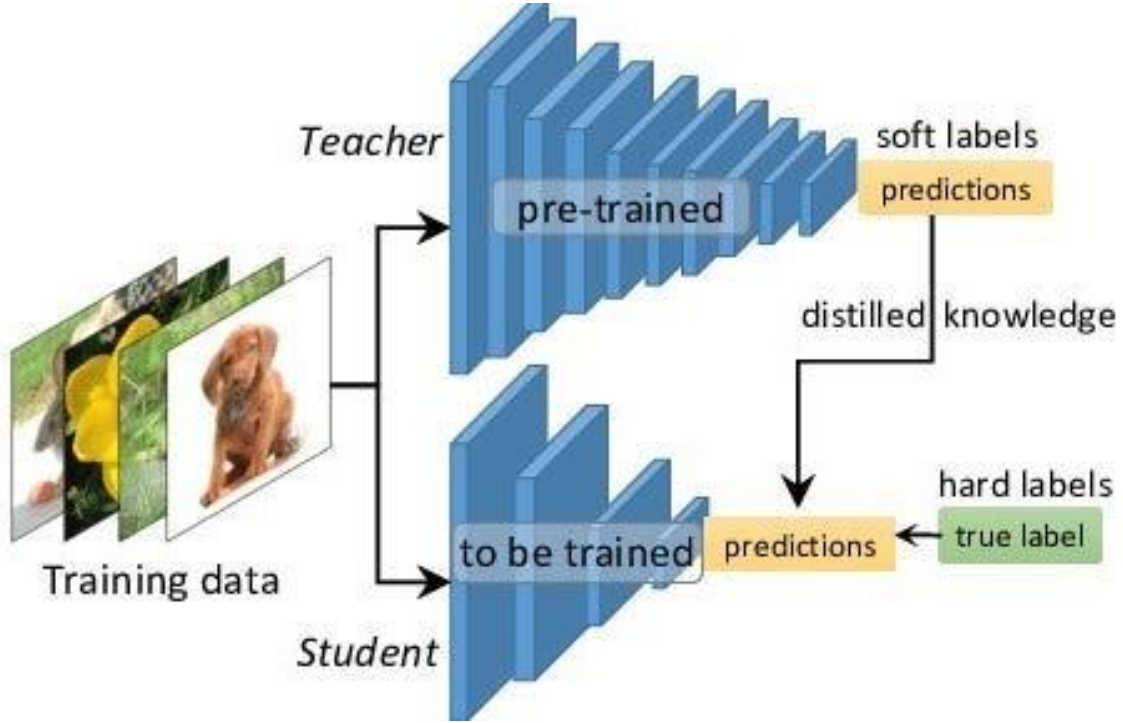
What happens with different architectures?

How might a student be "smarter"?

https://www.researchgate.net/figure/The-pseudo-code-of-knowledge-distillation-algorithm-Algorithm-3-Knowledge-Distillation_tbl2_385905148

© Copyright Deep Sentinel

“Soft Labels”

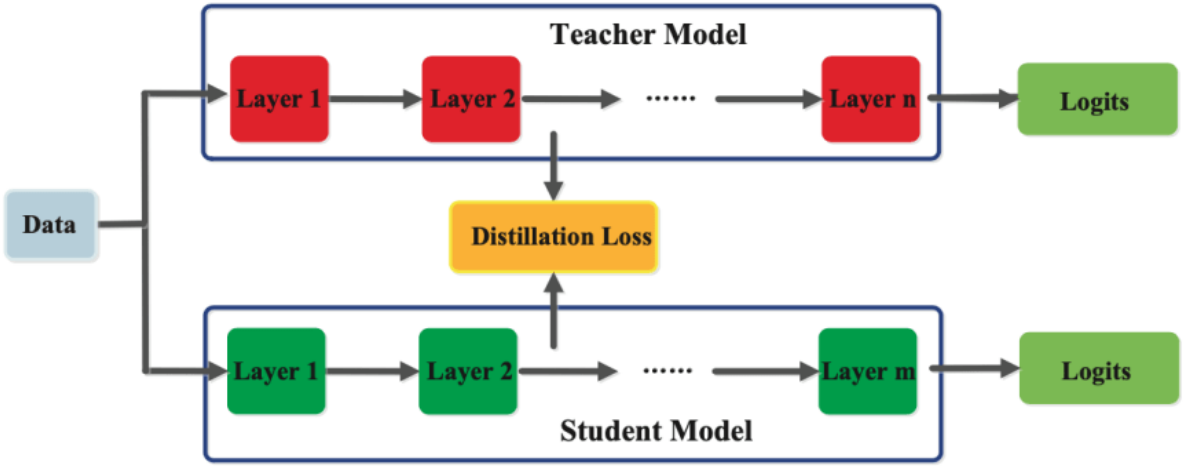


https://pub.towardsai.net/a-gentle-introduction-to-knowledge-distillation-6240bf8eb8ea

Deep Sentinel

© Copyright Deep Sentinel

Training: Feature-based Distillation



- *Latents* as Target: one or more feature layer
- Loss functions: (same approach)
  - Classification: Cross Entropy (standard classification loss) +
  - Distillation: KL Divergence (distillation difference)

https://neptune.ai/blog/knowledge-distillation

Deep Sentinel

© Copyright Deep Sentinel

Training Pseudocode

- for (*epochs*) do:
  - for each (x, y) ε D
    - $y_t = T(x)$ [Store Latent layers → L_t)
    - $y_s = S(x)$ [Store Latent layers → L_S)
    - Loss = Cross-Entropy Loss (y, y_s) + KL Divergence (L_t, L_s)
    - BackProp (S, Loss)

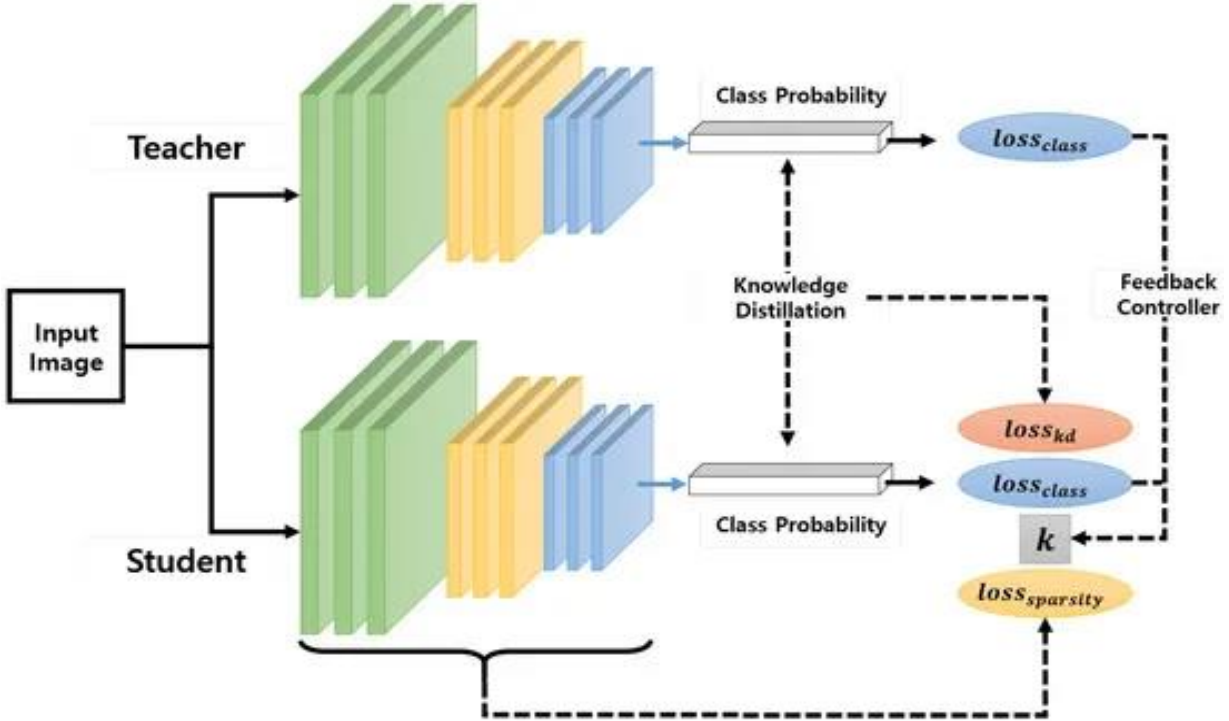
D = Dataset of target domain
T = Teacher network (forward compute to logits)
S = Student network

https://www.researchgate.net/figure/The-pseudo-code-of-knowledge-distillation-algorithm-Algorithm-3-Knowledge-Distillation_tbl2_385905148

Insights:
How might a student be "smarter"? Can we still use different architectures?

© Copyright Deep Sentinel

The Importance of Loss Functions



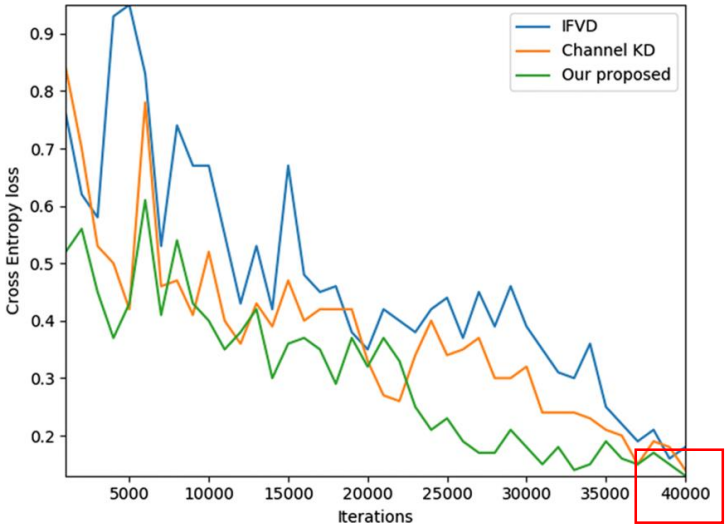
https://medium.com/analytics-vidhya/knowledge-distillation-for-convolutional-networks-e73682c611e8

© Copyright Deep Sentinel

Accuracies

- TLDR: Really good ☺

		CUB-200-2011 [40]				Cars 196 [14]				Stanford Online Products [21]			
	K	1	2	4	8	1	2	4	8	1	10	100	1000
GoogLeNet [35]	LiftedStruct [21]-128	47.2	58.9	70.2	80.2	49.0	60.3	72.1	81.5	62.1	79.8	91.3	97.4
	N-pairs [34]-64	51.0	63.3	74.3	83.2	71.1	79.7	86.5	91.6	67.7	83.8	93.0	97.8
	Angular [41]-512	54.7	66.3	76.0	83.9	71.4	81.4	87.5	92.1	70.9	85.0	93.5	98.0
	A-BIER [22]-512	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1	74.2	86.9	94.0	97.8
	ABE8 [13]-512	60.6	71.5	79.8	87.4	**85.2**	**90.5**	94.0	96.1	**76.3**	**88.4**	94.8	98.2
	RKD-DA-128	60.8	72.1	81.2	**89.2**	81.7	88.5	93.3	96.3	74.5	88.1	95.2	98.6
	RKD-DA-512	**61.4**	**73.0**	**81.9**	89.0	82.3	89.8	**94.2**	**96.6**	75.1	88.3	**95.2**	**98.7**
ResNet50 [10]	Margin [42]-128	63.6	74.4	83.1	90.0	79.6	86.5	91.9	95.1	72.7	86.2	93.8	98.0
	RKD-DA-128	**64.9**	**76.7**	**85.3**	**91.0**	**84.9**	**91.3**	**94.8**	**97.2**	**77.5**	**90.3**	**96.4**	**99.0**



https://cvlab.postech.ac.kr/research/RKD/

Deep Sentinel

© Copyright Deep Sentinel

Conclusion:
Distillation for Embedded Applications

- **Resource constrained:** SOC (resource conflict), FPGA (neurons/memory), NPU (compute)
- **Domain specific:** Domain-specific applications of devices enable highly-compressed/focused models
- **High quality:** DNNs are huge and generally sparse – compression is not lossless, but it can be very, very high quality; even better

© Copyright Deep Sentinel

Wait, How Do I Get Started???

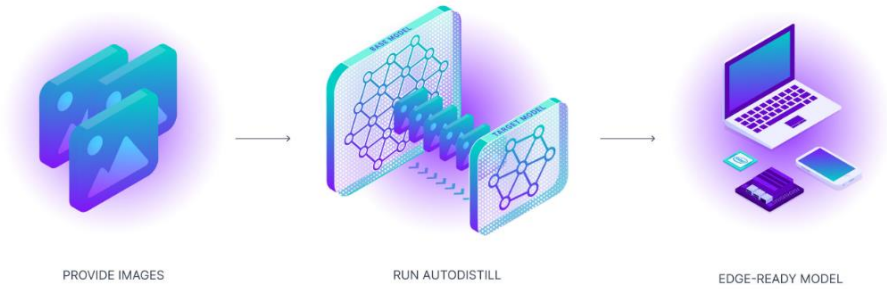
Don’t wait.
Try now, learn sooner.

autodistill

pypi package 0.1.29 | downloads 8.6k/month | license Apache-2.0 | python 3 | Open in Colab

Autodistill uses big, slower foundation models to train small, faster supervised models. Using `autodistill`, you can go from unlabeled images to inference on a custom model running at the edge with no human intervention in between.

You can use Autodistill on your own hardware, or use the Roboflow hosted version of Autodistill to label images in the cloud.



Deep Sentinel

© Copyright Deep Sentinel

Distillation

David Selinger

selly@deepsentinel.com

Deep Sentinel

© Copyright Deep Sentinel