ChatGPT

Proposed CIFAR-10 Configuration Permutations (≤400k Parameters)

Below we present a comprehensive set of JSON-style experiment configurations. These are grouped by **Model Variant 0–5**, each representing a distinct architecture. All configurations respect the **≤400,000** trainable parameter limit (for variants 0 and 1, we reduce the final dense layer to 64 units to meet this budget). We systematically vary **model depth/width, regularization, optimization, training schedule, data augmentation, early stopping**, etc., to explore their impact on accuracy vs. generalization. Each configuration can be injected into the pipeline by modifying the provided `default.json` template accordingly. In general, we follow best practices from the literature: e.g. **weight decay** (~5×10^-4^) and **dropout** (e.g. 0.5) as used in VGG-style networks [1] [2], **batch normalization** to stabilize deeper nets [3], **SGD with momentum** (0.9) or **Adam** optimizers [1], learning rate **scheduling** upon plateau [1], and **data augmentation** (random flips/crops) which is standard for CIFAR-10 [4]. We also employ **early stopping** in some runs to prevent over-training [5]. Any permutation that is theoretically well-founded (even marginal cases) is included, as instructed.

(Note: Omitted fields in tables are identical to `default.json`*. "Schedule" refers to a ReduceLROnPlateau scheduler with factor 0.5 and patience 3 by default, and "EarlyStop" uses patience 5 with restore-best-weights when enabled.)*

Variant 0 – Shallow VGG-style CNN (No BatchNorm)

A 2-block CNN with 32/64 filters and a Dense layer. Roughly VGG-like architecture but shallow [6]. Lacks batch normalization, so it is prone to overfitting and unstable with large learning rates. We experiment with strong regularization (L2 and dropout) as used in VGG [1] to tame its ~330k parameters, and include an unregularized baseline to illustrate overfitting.

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light Dataset
V0-Base	**Off**	**Off**	Adam (0.01)	20	32	Off	Off	Off	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light Dataset
V0-Base: No regularization baseline (no L2, no dropout, no augmentation) to gauge overfitting. With ~330k params and no BN, the model will likely memorize training data (overfit) [1] .									
V0-L2	**On** (5e-4)	Off	Adam (0.01)	20	32	Off	Off	Off	Off
V0-L2: L2 weight decay only (λ=5×10^-4^) [2] . Weight decay should help regularize the large fully-connected layer by penalizing weights [1] .									
V0-Drop	Off	**On** (0.5)	Adam (0.01)	20	32	Off	Off	Off	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light Dataset
V0-Drop: Dropout only (50% dropout) [1]. This tests dropout's effect in isolation – dropout (0.5) was critical in VGG's dense layers to prevent overfitting [2].									
V0-Reg	On (5e-4)	On (0.5)	Adam (0.001)	50	32	**On**	Off	**On**	Off
V0-Reg: Both L2 (5e-4) and dropout (0.5) enabled – a heavily regularized setup akin to VGG configurations [1]. We lower Adam's LR to 0.001 for stability (no BN). LR scheduler is active to halve LR on plateaus, allowing further refinement [1]. Data augmentation is used to improve generalization [4].									

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light Dataset
V0-Light	On (5e-4)	On (0.5)	Adam (0.001)	20	16	On	**On** (p3)	**On**	**On**
V0-Light: Regularized as above but on **Light Mode** (smaller dataset). Early stopping (patience 3) is added to halt training once validation stops improving [5], since with fewer samples the model may overfit quickly. Augmentation remains on to bolster the limited data.									

Rationale: Variant 0's simple VGG-like network (2×Conv -> Pool blocks, one dense layer) easily overfits CIFAR-10 without BN or regularization. The unregularized baseline (V0-Base) is expected to overfit severely (high train, poor val). Adding **weight decay** (V0-L2) or **dropout** (V0-Drop) individually should reduce overfitting modestly – weight decay constrains weight magnitude [1], while dropout randomly zeroes activations to prevent co-adaptation [2]. The combination (V0-Reg) should yield much better generalization [2], albeit training slower. **Data augmentation** further helps generalization by expanding effective data [4]. On the small "light" dataset, strong regularization plus **early stopping** is crucial to avoid memorization [5]. We use Adam for Variant 0 due to its faster convergence on unnormalized networks.

Variant 1 – VGG-style with BatchNorm

Same architecture as Variant 0 but with Batch Normalization layers after each convolution. This mirrors a VGG-like model with BN (improving training stability) [7] [8]. BN permits higher learning rates and often reduces the need for dropout [3]. We test both SGD (with momentum) and Adam, vary dropout rates, and compare no-reg vs reg.

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V1-Base	Off	Off	Adam (0.01)	20	32	Off	Off	Off	Off
V1-Base: No reg baseline **with BN**. BatchNorm stabilizes training even without dropout [3], so this may perform slightly better than V0-Base. Still, with ~330k params, overfitting is expected (BN alone can't fully prevent it [3]).									
V1-L2	On (5e-4)	Off	Adam (0.01)	20	32	Off	Off	Off	Off
V1-L2: Weight decay only (5e-4). Tests if L2 alone (with BN) sufficiently controls overfitting. VGG-style models typically use this weight decay by default [1].									
V1-Drop	Off	On (0.5)	Adam (0.01)	20	32	Off	Off	Off	Off
V1-Drop: Dropout only (0.5). With BN present, dropout's benefit vs. potential training slowdown can be observed. (ResNet authors often omit dropout when using BN [3], so BN+dropout is an open question.)									

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V1-SGD	On (5e-4)	On (0.5)	**SGD (0.1, 0.9)**	50	32	**On**	Off	On	Off
V1-SGD: BN-enabled network with classic **SGD+momentum 0.9** [1]. Initial LR 0.1 (as in many CIFAR trainings [9]) is used, with ReduceLROnPlateau scheduler to drop it on stagnation. This config emulates the training regime of VGG/ResNet (which used LR=0.1 with step-wise decay) [1] [9]. Weight decay 5e-4 and dropout 0.5 are enabled per VGG-like best practice [1].									
V1-Adam	On (5e-4)	On (0.5)	Adam (0.01)	20	32	On	Off	On	Off
V1-Adam: Same as V1-SGD but using **Adam** optimizer for a faster convergence test. We run 20 epochs with plateau scheduling – Adam (LR 0.01) typically reaches good accuracy quickly. This gauges if Adam attains similar generalization as SGD [1] on this BN model.									

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V1-Drop ↑	On (5e-4)	**On (0.7)**	SGD (0.1, 0.9)	50	32	On	Off	On	Off
V1-Drop ↑ : High dropout rate (70%) to test an extreme regularization case. Higher dropout can further reduce overfit but may hurt training if too excessive. Included to see if a very aggressive dropout yields any gain (as sometimes tried in literature).									
V1-Light	On (5e-4)	On (0.5)	Adam (0.001)	20	16	On	**On** (p5)	On	**On**
V1-Light: Variant 1 on the small dataset. We use strong regularization (L2+dropout) and **early stopping** (patience 5) to prevent overfitting the limited data. Adam with a lower LR (0.001) is used, as BN allows stable training even with few samples. Augmentation is on to maximize effective data.									

Rationale: Variant 1 adds **Batch Normalization**, which allows higher learning rates and often improves generalization for deep CNNs [3]. We demonstrate that with BN, an aggressive SGD schedule (V1-SGD) can be used – mirroring the training approach in VGG/ResNet (momentum 0.9, LR ~0.1 with decay) [1] [9]. We expect V1-SGD to reach strong performance, as BN+weight decay was key to ResNet's success without needing dropout [3]. We include Adam runs (V1-Adam) to compare optimizer efficacy: Adam may converge faster but sometimes yields slightly less final accuracy than tuned SGD [1]. The **dropout rate** sweep (0.5 vs

0.7) gauges if more dropout helps – likely 0.7 is overkill, but we include it as a marginal case. The Light-mode experiment shows if BN+reg can handle extremely limited data (with early stopping to avoid overfitting [5]).

Variant 2 – Narrow CNN, Global Average Pooling

A smaller CNN: 2 conv blocks with fewer filters (16 and 32), no dense layer (Global Average Pooling + softmax output). Only ~17k parameters – significantly under the 400k cap. This model has limited capacity (risk of underfitting). We examine minimal vs added regularization, and the effect of augmentation.

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V2-Base	Off	Off	Adam (0.01)	50	32	On	Off	**Off**	Off
V2-Base: No regularizers, letting the small model fully utilize its capacity. With so few parameters, overfitting is less likely – this baseline checks if it can learn the data patterns at all. (Scheduler is on to help fine-tune as training plateaus.)									
V2-Aug	Off	Off	Adam (0.01)	50	32	On	Off	**On**	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V2-Aug: Same as V2-Base but with data augmentation. This reveals if augmentation helps a low-capacity model. (Augmenting might slightly increase the effective data and improve test accuracy [2], or it could make learning harder given the model's limited capacity.)									
V2-L2	**On** (5e-4)	Off	Adam (0.01)	50	32	On	Off	Off	Off
V2-L2: Weight decay added. Even though the model is small, a bit of L2 (5e-4) can improve generalization by smoothing weights [2]. We verify if it helps or simply slows learning (since underfitting might be a concern).									

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V2-Drop	Off	**On (0.2)**	Adam (0.01)	50	32	On	Off	Off	Off
V2-Drop: Dropout (20%) added alone. We choose a mild rate (0.2) because heavy dropout on such a small network could remove too many features. This tests if even light dropout aids generalization (small models sometimes don't need it).									
V2-Reg	On (5e-4)	On (0.2)	Adam (0.01)	50	32	On	Off	On	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V2-Reg: Both L2 and dropout (0.2) with augmentation – a fully regularized scenario for the tiny model. This might be over-regularized (given the model's limited capacity), but we include it to see if combining all techniques yields any marginal benefit.									

Rationale: Variant 2 is extremely lightweight (only 16–32 filters, no dense). Its baseline (V2-Base) may actually *underfit* CIFAR-10, so regularization might not be necessary. We expect **no-reg** to maximize its reachable accuracy. Data **augmentation** (V2-Aug) could modestly help generalization even for a small model, though if the model can't even fit the original data well, augmentation might not yield much gain. Adding **L2 or dropout** (V2-L2, V2-Drop) might actually slightly hurt training if the model is capacity-limited – these are included to confirm whether any regularization is beneficial in this regime. The fully regularized case (V2-Reg) is likely overkill, but we include it as a stress test of "all generalization techniques" on a small network. Overall, this variant probes the lower bound of model complexity: with only ~17k params, it should be far from overfitting, so it emphasizes how much accuracy can be obtained under strict capacity limits.

Variant 3 – Moderate CNN with BN, Global Pool

This is like a "mini-VGG BN" model: 32 and 64 filters (same as V1) with BN, but no dense layer (uses Global Average Pooling). ~66k parameters. It's a balanced model – larger than Var.2 but still modest. We vary regularization and optimizer to see what yields the best trade-off.

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V3-Base	Off	Off	Adam (0.01)	50	32	On	Off	On	Off
V3-Base: Baseline with BN, no dense. With ~66k params, some overfitting is possible but less severe than Var.1. We use Adam and augmentation; no explicit reg so we can observe BN's inherent regularization effect.									
V3-L2	**On** (5e-4)	Off	Adam (0.01)	50	32	On	Off	On	Off
V3-L2: Weight decay only. Likely improves generalization a bit (ResNet on CIFAR-10 used L2=1e-4 without dropout for optimal results [3]). Here we use 5e-4 similar to VGG/ResNet settings [1].									
V3-Drop	Off	**On (0.5)**	Adam (0.01)	50	32	On	Off	On	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V3-Drop: Dropout only (0.5). With no dense layer, dropout applies to the GAP output (64-d). This tests if dropout helps BN networks of moderate size.									
V3-Reg	On (5e-4)	On (0.5)	Adam (0.01)	50	32	On	Off	On	Off
V3-Reg: L2 + dropout together (and augmentation) – the full regularization suite. We expect this to yield the best generalization for variant3 (at the cost of some training speed).									
V3-SGD	On (5e-4)	On (0.5)	**SGD (0.1, 0.9)**	50	32	On	Off	On	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V3-SGD: Same reg settings as V3-Reg but using SGD w/ momentum 0.9 and LR 0.1. BN should allow this high learning rate without issues [3]. This will show if a tuned SGD can outperform Adam on this architecture in final accuracy (as often seen in vision models) [1].									

Rationale: Variant 3 (BN + GAP) strikes a middle ground: it has enough capacity to potentially overfit, but BN and the absence of a huge dense layer already mitigate that. The baseline with **no reg** (V3-Base) with augmentation might already generalize decently. Adding **weight decay** (V3-L2) is expected to help, as weight decay was beneficial even for ResNets on CIFAR [3]. **Dropout** (V3-Drop) may or may not help here – ResNet results suggest dropout wasn't needed with BN [3], but our network is smaller, so dropout could still add robustness. The fully regularized config (V3-Reg) should be very resilient to overfitting and likely achieves the best test accuracy. We compare **Adam vs SGD** (V3-Reg vs V3-SGD): with BN, SGD+momentum at LR 0.1 (with schedule) often eventually surpasses Adam's performance on image tasks [1]. This will validate if the theoretical generalization advantage of SGD holds for a moderate-sized BN model.

Variant 4 – Depthwise Separable CNN (MobileNet-like)

A MobileNet-inspired architecture [10] with depthwise separable convolutions. Two blocks: 32 filters then 64 filters (pointwise conv outputs) with depthwise conv in between. Uses BN and GAP. Extremely small (~9.8k params), well under 400k. We explore if scaling regularization back (or up) affects this highly efficient model, and consider using more capacity.

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V4-Base	Off	Off	Adam (0.01)	50	32	On	Off	Off	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V4-Base: No regularization baseline. This MobileNet-like model has so few parameters that it might underfit; we start with no reg to let it learn freely.									
V4-Aug	Off	Off	Adam (0.01)	50	32	On	Off	**On**	Off
V4-Aug: Adding data augmentation. With such low capacity, augmentation could either help slightly in generalization or make fitting harder – we test its effect [2].									
V4-L2	**On** (5e-4)	Off	Adam (0.01)	50	32	On	Off	On	Off
V4-L2: Weight decay enabled. Regularizing might not be necessary for only ~10k weights, but a small L2 could improve stability of learning for depthwise filters.									

Config ID	L2 (λ)	Dropout (rate)	Optimizer	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V4-Drop	Off	**On (0.2)**	Adam (0.01)	50	32	On	Off	On	Off
V4-Drop: Dropout 20% enabled. Depthwise conv nets (with BN) typically did not use dropout in the original MobileNet paper, focusing on weight decay instead [3]. We try a light dropout to see if it has any effect.									
V4-Reg	On (5e-4)	On (0.2)	Adam (0.01)	50	32	On	Off	On	Off
V4-Reg: L2 and dropout combined (plus augmentation). This is likely over-regularized given the model's tiny size, but we include it for completeness (to see if even a 10k-param model can benefit from all techniques).									

Rationale: Variant 4 is a **MobileNet-like** efficient model. Its depthwise separable layers dramatically cut parameter count (conv parameters are ~1% of a regular conv of equal dimensions [10]), resulting in only ~9.8k params. This model is so small that overfitting is not the main issue – rather, *underfitting* could be. Therefore, heavy regularization isn't needed. We anticipate that the **baseline** (V4-Base) might already not

reach very high accuracy due to limited capacity. **Augmentation** (V4-Aug) might improve test accuracy slightly by providing more variety [2], but the gain could be limited by the model's representational power. Adding **weight decay or dropout** (V4-L2, V4-Drop) on top of BN likely yields no benefit or even hurts (the network might need to use all its small capacity). These are included to verify that hypothesis. The fully regularized case (V4-Reg) is mainly to observe if combining reg methods on an extremely small model can ever be beneficial – we expect not, but include the experiment. *Note:* If one wanted to utilize more of the 400k budget with this architecture, one could increase its **width multiplier** (e.g. 2× filters would still be ≪400k params) [11] – however, here we stick to the given 32/64 filters to focus on the baseline MobileNet design.

Variant 5 – Residual CNN (ResNet-style)

A small ResNet-inspired model: 2 conv blocks (32 and 64 filters) with identity shortcuts (1×1 conv projections) and BN, followed by GAP. ~68k params. Residual connections improve training of deeper networks [12] [13]; here the network isn't very deep, but skip connections may still aid generalization. We try weight decay (as ResNet did) vs dropout, and SGD vs Adam.

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V5-Base	Off	Off	Adam (0.01)	50	32	On	Off	**Off**	Off
V5-Base: Baseline residual network with no explicit reg (no L2, no dropout, no aug). Thanks to BN and skip connections, it should train stably [3]. This config examines how well the raw capacity ~68k can fit CIFAR-10.									
V5-Aug	Off	Off	Adam (0.01)	50	32	On	Off	**On**	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V5-Aug: Baseline + data augmentation. ResNet experiments on CIFAR-10 universally employed augmentation (random crop/ flip) to boost performance [4] – we expect noticeable gains in test accuracy here as well.									
V5-L2	**On** (1e-4)	Off	**SGD (0.1, 0.9)**	50	32	**On**	Off	On	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V5-L2: Weight decay only, λ=1×10^-4^ (we use the value from He et al.'s ResNet paper [3]). **SGD** with momentum and initial LR 0.1 is used, mimicking ResNet's original training regime on CIFAR [3] [9]. No dropout (ResNet did not use dropout) [3]. This should yield strong performance if ResNet theory holds.									
V5-Drop	Off	**On (0.5)**	SGD (0.1, 0.9)	50	32	On	Off	On	Off

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V5-Drop: Dropout only (0.5) with SGD. This tests the effect of dropout **instead of** weight decay on a residual network. Given ResNet avoided dropout in favor of BN+L2 [3], we expect this to possibly underperform V5-L2 – included for comparison.									
V5-Reg	On (1e-4)	On (0.5)	SGD (0.1, 0.9)	50	32	On	Off	On	Off
V5-Reg: Both weight decay (1e-4) and dropout (0.5) with SGD. This is a “belt-and-suspenders” approach – likely unnecessary per ResNet results, but we include it to observe if dropout adds anything extra on top of L2 in a residual model.									

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V5-Adam	On (5e-4)	Off	**Adam (0.001)**	20	32	On	Off	On	Off
V5-Adam: Use Adam optimizer (LR 0.001) with weight decay. Adam converges quickly; we run 20 epochs. This checks if Adam can reach similar accuracy on ResNet variant as SGD (possibly with slightly lower final generalization) [1]. We keep dropout off (since L2+BN should suffice).									
V5-Light	On (5e-4)	On (0.5)	Adam (0.001)	20	16	On	**On** (p5)	On	**On**

Config ID	L2 (λ)	Dropout (rate)	Optimizer (LR,mom)	Epochs	Batch	Schedule	EarlyStop	Augment	Light
V5-Light: Residual model on the light dataset. We apply moderate reg (L2 + dropout) and early stopping (patience 5) to curb overfitting. Adam is used for faster convergence on the small data. Augmentation is on to expand the tiny training set [2].									

Rationale: Variant 5 is inspired by **ResNet**. Key insights from ResNet paper: **identity shortcuts** allow deeper models to train effectively and often **eliminate the need for dropout** when combined with BN and weight decay [3]. In our two-block variant, depth is limited, but the residual connections may still improve learning of features. We follow ResNet's recommended regularization: **weight decay 1e-4, no dropout** (V5-L2) – this configuration should perform strongly, as evidenced by ResNet-20's success on CIFAR-10 with those settings [3]. We compare to a **dropout-only** run (V5-Drop) to see if dropout can substitute for weight decay; likely, weight decay yields better calibrated networks [3]. The combined reg (V5-Reg) will show if adding dropout on top of L2 has any marginal benefit (ResNet results suggest not much [3]). We also include an **Adam**-optimized run (V5-Adam) to evaluate optimizer differences on a ResNet-style model – SGD with momentum was traditionally preferred for ResNets [3], but Adam might reach respectable results with less tuning. Finally, the Light dataset case (V5-Light) tests if a residual network can generalize from very limited data: we expect early stopping to kick in before overfitting [5], and the residual connections + heavy regularization should help retain generalization despite the data paucity.

References: The configuration choices are guided by the cited works. **VGG**-style networks emphasize deeper layers with small filters and use dropout (0.5) and weight decay (5×10^-4^) to combat overfitting [1] [2]. **ResNet** introduced residual connections which, along with batch normalization and weight decay (~1×10^-4^), enabled training of very deep nets without dropout [3]. Our variant5 mirrors those principles on a smaller scale. **MobileNet** demonstrated that depthwise separable convolutions drastically reduce parameters with minimal loss in accuracy [10]; we leverage that efficiency in variant4. We also incorporate standard practices like data augmentation (random crops/horizontal flips) for CIFAR-10 [4] and learning-

rate scheduling (reduce on plateau) to improve convergence [1]. **Early stopping** is used in some cases as a regularization technique to prevent over-training on limited data [5]. Each configuration above represents a plausible and "promising" experiment setting, covering a wide spectrum from highly regularized to minimally regularized, and from low-capacity models to near-budget models – all within the 400k parameter limit as required.

---

[1] [6] CNN Architectures — VGGNet. Small filters, deeper network. | by Gary(Chang, Chih-Chun) | Deep Learning#g | Medium
https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

[2] [PDF] Explicit Regularisation, Sharpness and Calibration - OpenReview
https://openreview.net/pdf?id=ZQTiGcykl6

[3] [4] [9] arxiv.org
https://arxiv.org/pdf/1512.03385

[5] Early Stopping in Deep Learning: A Simple Guide to Prevent Overfitting | by Piyush Kashyap | Medium
https://medium.com/@piyushkashyap045/early-stopping-in-deep-learning-a-simple-guide-to-prevent-overfitting-1073f56b493e

[7] [8] [12] [13] model.py
file://file-ENu9aZUt5tG1nkMtf5QnYc

[10] [11] task-001_1704.04861v1.pdf
file://file-2TyaHoXc6MbhmUPEKQD5di