INDEX

A
active learning, 195, 203–204, 222
Adam optimizer, 42, 73, 211, 212–213
adapter methods, 119, 121–123,
125–126, 216
Add & Norm step, 107
adversarial examples, 27
adversarial validation, 154, 190–191,
217, 221
AI (artificial intelligence)
data-centric, 143–146, 217
model-centric, 143–144, 145
AlexNet, 6, 7
asymptotic coverage guarantees,
confidence intervals, 177
attention mechanism. See also
self-attention mechanism
Bahdanau, 99–101, 103, 112
transformers, 40, 43–45, 46, 47
augmented data
reducing overfitting with, 24–25,
26, 210
for text, 93–97, 214–215
autoencoders
defined, 51
latent space, 5–6
variational, 51–52
automatic prompt engineering
method, 125
autoregressive decoding, 107–110
autoregressive models, 54–55, 57, 63–64
auxiliary tasks, 199
Bahdanau attention mechanism,
99–101, 103, 112
BART encoder-decoder architecture, 112
base classes in few-shot learning, 16
Basic Linear Algebra Subprograms
(BLAS), 148, 152
batched inference, 147–148
batch normalization (BatchNorm),
73, 213
Berry–Esseen theorem, 167, 171
BERT model
adopting for classification tasks,
112, 215–216
distributional hypothesis, 91, 92
encoder-only architectures,
107–108
BERTScore, 132–133, 134, 216–217
bias units
in convolutional layers, 70–71
in fully connected layers, 72, 76
binomial proportion confidence
interval, 171
BLAS (Basic Linear Algebra
Subprograms), 148, 152
BLEU (bilingual evaluation
understudy) score, 128,
129–131, 133, 134
bootstrapping
improving performance with
limited data, 194
out-of-bag, 167–169, 170, 171
test set predictions, 169, 170, 171, 219
C
 calibration set, 176
CBO (continuous bag-of-words)
 approach, 90
CE (cross-entropy) loss, 128, 182
central limit theorem, 167, 171
ChatGPT model
autoregressive models, 54
randomness by design, 63
reinforcement learning with
human feedback (RLHF), 124
stateless vs. stateful training, 141, 217
zero-shot learning, 196
classic bias-variance theory, 31, 35
classification tasks
adopting encoder-style transformers for, 112,
215–216
cross entropy and, 128
fine-tuning decoder-style transformers for, 112, 216
using pretrained transformers for, 113–116
Cleanlab open source library, 146
CNNs. See convolutional neural networks; neural networks
coloring video data, 208
Colossal AI, 37, 42
COMET neural framework, 131, 135
computer vision
calculating number of parameters, 69–73, 212–213
distributional hypothesis, 214
fully connected and convolutional layers, 75–78, 213
large training sets for vision transformers, 79–85,
213–214
self-attention mechanism, 103, 215
concept drift, 155–156
confidence intervals
asymptotic coverage guarantees, 177
bootstrapping test set predictions, 169
bootstrapping training sets, 167–169
vs. conformal predictions, 173–178
defined, 164–165
normal approximation intervals, 166–167
overview, 163, 173
and prediction intervals, 174
recommendations for, 170, 178
retraining models with different random seeds, 169–170
confidence scores in active learning, 204, 222
conformal predictions
benefits of, 177–178
computing, 175–176
example of, 176–177
overview, 173
and prediction intervals, 174
recommendations for, 178
connectivity, 80, 81
consistency models, 56–57, 58, 212
continuous bag-of-words (CBOW) approach, 90
contrastive learning, 208
contrastive self-supervised learning, 12–14
convolutional layers
calculating number of parameters in, 70–71
as high-pass and low-pass filters, 84
recommendations for, 78
replacing fully connected layers with, 75–78
convolutional neural networks (CNNs).
See also neural networks
calculating number of parameters in, 69–73, 212–213
embeddings from, 4, 6, 207
high-pass and low-pass filters in, 84
inductive biases in, 80–82
recommendations for, 84
with vision transformers, 79, 82–83, 84
convolution operation, 61–62
cosine similarity, 132, 134, 216
count data, 161
covariate shift, 153–154, 156, 157
CPUs, data parallelism on, 42, 211
cross-entropy (CE) loss, 128, 182
cross-validation
5-fold cross-validation, 187, 188, 221
k-fold cross-validation, 185–188, 221
leave-one-out cross-validation (LOOCV), 188, 221
10-fold cross-validation, 187, 188
CUDA Deep Neural Network library (cuDNN), 62

data. See also limited labeled data
applying self-supervised learning to video, 14, 208
count, 161
reducing overfitting with, 23–27, 209–210
self-supervised learning for tabular, 14, 208
synthetic, generation of, 96–97
unlabeled, in self-supervised learning, 10, 11
data augmentation
to reduce overfitting, 24–25, 26, 210
for text, 93–97, 214–215
data-centric AI, 143–146, 217
data distribution shifts
covariate shift, 153–154
domain shift, 155–156
label shift, 154–155
overview, 153
types of, 156–157
data parallelism, 37, 38, 39–40, 41–42, 211
datasets
for few-shot learning, 15
sampling and shuffling as source of randomness, 60
for transformers, 45
DBMs (deep Boltzmann machines), 50–51, 57
dead neurons, 209
decision trees, 204
decoder network (VAE model), 51–52
decoders
in Bahdanau attention mechanism, 100–101
in original transformer architecture, 105–106, 107
decoder-style transformers. See also encoder-style transformers
contemporary transformer models, 111–112
distributional hypothesis, 91
encoder-decoder hypothesis, 110 overview, 105, 108–110
synthetic data generation, 96–97
terminology related to, 110
deep Boltzmann machines (DBMs), 50–51, 57
deep generative models.
See generative AI models
deep learning. See also generative AI models
embeddings, 3–7, 207
few-shot learning, 15–18, 208–209
lottery ticket hypothesis, 19–21, 209
multi-GPU training paradigms, 37–42, 211
reducing overfitting with data, 23–27, 209–210
with model modifications, 29–36, 210
self-supervised learning, 9–14, 208
sources of randomness, 59–65, 212
transformers, success of, 43–47, 211
DeepSpeed, 37, 42
deletion, word, as data augmentation technique, 94
deterministic algorithms, 62, 65
diffusion models, 55–56, 57, 58
dimension contrastive self-supervised learning, 14
direct convolution, 61
discriminative models, 49–50
discriminator in GANs, 52–53
distance, embeddings as encoding, 5
distance functions, 179–183
distributional hypothesis, 89–92, 214
domain shift (joint distribution shift), 155–156, 157
double descent, 32–33, 36
downstream model for pretrained transformers, 114
downstream task, 11
drivers as source of randomness, 62
dropout, 30, 36, 61, 64–65, 212
E
early stopping, 30–31, 35, 210
EBMs (energy-based models), 50–51
EfficientNetV2 CNN architecture, 85
embeddings
distributional hypothesis, 90–91
in few-shot learning, 17
latent space, 5–6
in original transformer
architecture, 106
overview, 3–5
representations, 6
emergent properties, GPT models, 110
coder-decoder models, 110, 111
coder network (VAE model), 51–52
coder
in Bahdanau attention
mechanism, 100–101
in original transformer
architecture, 105–107
encoder-style transformers. See also
decoder-style transformers
contemporary transformer models,
111–112
encoder-decoder hybrids, 110
overview, 105, 107–108
terminology related to, 110
energy-based models (EBMs), 50–51
ensemble methods, 33–34, 35, 210,
221, 222
episodes in few-shot learning, 16
Euclidean distance, 181
evaluation metrics for generative LLMs
BERTScore, 132–133
BLEU score, 129–131
overview, 127–128
perplexity, 128–129
ROUGE score, 131–132
surrogate metrics, 133
extrinsic metrics, 128
f
few-shot learning. See also
in-context learning
datasets and terminology, 15–17
limited labeled data, 195–196, 203
overview, 15
reducing overfitting with, 25
FFT (fast Fourier transform)-based
convolution, 62, 65
fine-tuning pretrained transformers,
113–116, 119–124,
125–126, 216
finite-sample guarantees of conformal
predictions, 177
5-fold cross-validation, 187, 188, 221
flow-based models (normalizing flows),
53–54, 57
Fréchet inception distance
approach, 212
fully connected (FC) layers
calculating number of parameters
in, 70, 72
lack of spatial invariance or
equivariance, 82
recommendations for, 78
replacing with convolutional
layers, 75–78
using to create embeddings, 6, 207
G
generalization accuracy, 164
generalization performance, 32–33
generative adversarial networks
(GANs), 52–53, 54, 57, 58
generative AI models
autoregressive models, 54–55
consistency models, 56–57
diffusion models, 55–56
energy-based models, 50–51
flow-based models, 53–54
generative adversarial networks,
52–53
generative vs. discriminative
modeling, 49–50
overview, 49
randomness and, 62–64
recommendations for, 57
variational autoencoders, 51–52
F
fast Fourier transform (FFT)-based
convolution, 62, 65
FC layers. See fully connected layers
feature selection, self-attention as
form of, 46, 211
generative large language models. See evaluation metrics for generative LLMs; large language models; natural language processing
generator in GANs, 52–53
Gibbs sampling, 51
GPT (generative pretrained transformer) models
decoder-style transformers, 91, 109–110
fine-tuning for classification, 112, 216
randomness by design, 63
self-prediction, 12
GPUs. See multi-GPU training paradigms
grokking, 32–33, 36

H
hard attention, 211
hard parameter sharing, 200
hard prompt tuning, 117–118
hardware as source of randomness, 62
hierarchical processing in CNNs, 80
histograms, 207
holdout validation as source of randomness, 60
homophones, 92, 214
human feedback, reinforcement learning with, 124
hyperparameter tuning, 188

I
image denoising, 56–57
image generation, 51, 52, 54–57, 211–212
image histograms, 207
“An Image Is Worth 16x16 Words” (Dosovitskiy et al.), 83, 85
ImageNet dataset, 9, 14, 175
image processing.
See computer vision
importance weighting, 154, 155, 157, 218
in-context learning, 113, 116–119, 125, 216. See also few-shot learning
indexing, 118–119, 125
inductive biases
in convolutional neural networks, 80–82
limited labeled data, 202
overview, 79
in vision transformers, 83–84
inference, speeding up. See model inference, speeding up
inpainting, 194–195, 208
input channels in convolutional layers, 70–71, 76–77
input embedding, 4
input representations, 6, 207
InstructGPT model, 124, 126, 133, 135
inter-op parallelism (model parallelism), 37, 38, 39–40, 41–42
intra-op parallelism (tensor parallelism), 37, 38–40, 41–42, 211
intrinsic metrics, 128
iterative pruning, 20, 31

J
joint distribution shift (domain shift), 155–156, 157

K
kernel size in convolutional layers, 70–71, 76–77
k-fold cross-validation
determining appropriate values for k, 187–188
ensemble approach, 33–34
overview, 185–186
as source of randomness, 60
trade-offs in selecting values for k, 186–187
knowledge distillation, 31–33, 35, 36, 151, 199
Kullback–Leibler divergence (KL divergence), 32, 52, 211

L
L2 distance, 181
L2 regularization, 30, 35
labeled data, limited. See limited labeled data
label shift (prior probability shift), 154–155, 156
label smoothing, 27
language transformers. See transformers
large language models (LLMs).
See also natural language processing; transformers
distributional hypothesis, 91
evaluation metrics for, 127–135, 216–217
stateless vs. stateful training, 141, 217
synthetic data generation, 96–97
latent space, 3, 5–7
layer input normalization techniques, 34–35
layers
convolutional layers
calculating number of parameters in, 70–71
as high-pass and low-pass filters, 84
recommendations for, 78
replacing fully connected layers with, 75–78
normalization in original transformer architecture, 106–107
updating when fine-tuning pretrained transformers, 115–116
using to create embeddings, 207
leave-one-out cross-validation (LOOCV), 188, 221
limited labeled data
active learning, 195
bootstrapping data, 194
few-shot learning, 195–196
inductive biases, 202
labeling more data, 193–194
meta-learning, 196–197
multimodal learning, 200–202
multi-task learning, 199–200
overview, 193
recommendations for choosing technique, 202–203
self-supervised learning, 194–195
self-training, 199
semi-supervised learning, 198–199
transfer learning, 194
weakly supervised learning, 197–198
linear classifiers, 114
LLMs. See large language models;
natural language processing;
transformers
local connectivity in CNNs, 80, 81
logistic regression classifier, 49–50
LOOCV (leave-one-out cross-validation), 188, 221
loop fusion (operator fusion), 150–151
loop tiling (loop nest optimization), 149–150, 151, 152, 218
LoRA (low-rank adaptation), 119, 123–124, 125, 126, 216
loss function, VAEs, 52
lottery ticket hypothesis
overview, 19
practical implications and limitations, 20–21
training procedure for, 19–20
low-rank adaptation (LoRA), 119, 123–124, 125, 126, 216
low-rank transformation, 123
M
MAE (mean absolute error), 183, 220–221
majority voting, 33
MAPIE library, 178
masked (missing) input self-prediction methods, 12
masked frames, predicting, 208
masked language modeling, 91, 107–108, 194
mean absolute error (MAE), 183, 220–221
mean squared error (MSE) loss, 180–181
memory complexity of self-attention, 103, 215
metadata (meta-features) extraction, 197
meta-learning, 17, 196–197
METEOR metric, 131, 134
metrics, proper. See proper metrics
missing (masked) input self-prediction methods, 12
missing frames, predicting, 208
Mixup, 27
MLPs (multilayer perceptrons), 50, 82
MNIST dataset, 15, 18, 26, 208, 210
model-centric AI, 143–144, 145
model ensembling, 33–34, 35, 210, 221, 222
model evaluation. See predictive performance and model evaluation
model inference, speeding up
loop tiling, 149–150
operator fusion, 150–151
overview, 147
parallelization, 147–148
quantization, 151
vectorization, 148–149
model modifications, reducing overfitting with, 29–36, 210
model parallelism (inter-op parallelism), 37, 38, 39–40, 41–42
model weight initialization as source of randomness, 59–60
MSE (mean squared error) loss, 180–181
multi-GPU training paradigms
data parallelism, 38
model parallelism, 38
overview, 37
pipeline parallelism, 40
recommendations for, 41–42
sequence parallelism, 40–41
speeding up inference, 152, 218
tensor parallelism, 38–40
multilayer perceptrons (MLPs), 50, 82
multimodal learning, 200–202, 204
multi-task learning, 199–200, 204

N
naïve Bayes classifier, 49–50
natural language processing (NLP). See also transformers
data augmentation for text, 93–97, 214–215
distributional hypothesis, 89–92, 214
evaluating generative LLMs, 127–135, 211–212
self-attention, 99–103, 215
neural networks. See also convolutional neural networks; generative AI models; transformers
attention mechanism for, 99–101
calculating number of parameters in, 69–73, 212–213
embeddings, 3–7, 207
few-shot learning, 15–18, 208–209
lottery ticket hypothesis, 19–21, 209
multi-GPU training paradigms, 37–42, 211
reducing overfitting with data, 23–27, 209–210
with model modifications, 29–36, 210
self-attention, 99–103
self-supervised learning, 9–14, 208
sources of randomness, 59–65, 212
transformers, success of, 43–47, 211
next-sentence/next-word prediction task, 12, 108, 109, 194
NICE (non-linear independent components estimation), 53–54, 58
NLP. See natural language processing; transformers
noise
consistency models and, 56–57
diffusion models and, 56
noise injection, 95–96
nonconformity measure, 176–177
nondeterministic algorithms, 61
non-linear independent components estimation (NICE), 53–54, 58
normal approximation intervals, 166–167, 170, 171
normalizing flows (flow-based models), 53–54, 57
nucleus sampling (top-\(p\) sampling), 63–64, 212
NVIDIA graphics cards, 62, 65
N-way K-shot (few-shot learning), 15–16

O
ODE (ordinary differential equation) trajectory, 56–57
one-hot encoding, 4, 207
online resources, xxviii
operator fusion (loop fusion), 150–151
ordinal regression, 161–162, 218–219
ordinary differential equation (ODE) trajectory, 56–57
outlier detection, 218
out-of-bag bootstrapping, 167–169, 170, 171
output channels in convolutional layers, 70–71, 76–77
output layers, updating, 115–116
overfitting
overview, 23
reducing with data, 23–27, 209–210
reducing with model modifications, 29–36, 210
predictive performance and model evaluation. See also limited labeled data
confidence intervals vs. conformal predictions, 173–178, 219–220
constructing confidence intervals, 163–171, 219
k-fold cross-validation, 185–188, 221
Poisson and ordinal regression, 161–162, 218–219
proper metrics, 179–183, 220–221
training and test set discordance, 189–191, 221
prefix tuning, 119, 120–121, 125, 126, 216
pretext tasks, 10
pretrained transformers
adapting, 124–125
classification tasks, 113–116
in-context learning, indexing, and prompt tuning, 116–119
overview, 113
parameter-efficient fine-tuning, 119–124, 125, 126
reinforcement learning with human feedback (RLHF), 124
pretraining
encoder-only architectures, 107–108
to reduce overfitting, 25
with self-supervised learning, 10–11
with transfer learning, 9–10
transformers, via self-supervised learning, 45
for vision transformers, 83
prior probability shift (label shift), 154–155, 156
production and deployment
data distribution shifts, 153–157, 218
model inference, speeding up, 147–152, 218
stateless and stateful training, 139–141, 217
prompt tuning, 117–118
P
parallelization
model inference, speeding up, 147–148
of transformers, 45–46
parameter-efficient fine-tuning, 113, 119–124, 125, 126
parameters
calculating number of in CNNs, 69–73, 212–213
of transformers, scale and number of, 45, 47
patchifying inductive bias, 83, 85, 213–214
perplexity metric, 127–129
pipeline parallelism, 37, 40, 41, 42
PixelCNN model, 54, 58
pixel generation, autoregressive, 54–55
Poisson regression, 161–162, 218–219
polysemous words, 90
population parameters, 164
positive-unlabeled learning (PU-learning), 198
post-training quantization, 151
prediction intervals, 173–175, 178
prediction regions, 174–175
prediction sets, 174, 178, 219–220
predictive analytics in healthcare, 146, 217
proper metrics
  criteria for, 179–180
cross-entropy loss, 182
mean squared error loss, 180–181
overview, 179
protein modeling, 214
proximal policy optimization, 124, 126
pruning, 31, 32–33, 35, 36, 151
pseudo-labelers, 199
PU-learning (positive-unlabeled learning), 198
PyTorch framework, 59, 61, 62, 65, 149
Q
quantization, 151, 152
quantization-aware training, 151
R
random characters, 95
random initialization, 209
randomness, sources of
dataset sampling and shuffling, 60
different runtime algorithms, 61–62
and generative AI, 62–64
hardware and drivers, 62
model weight initialization, 59–60
nondeterministic algorithms, 61
overview, 59
random seeds, 169–170
recall-oriented understudy for gisting evaluation (ROUGE) score, 128, 131–132, 133, 134
reconstruction error, measuring, 218
reconstruction loss, 52
rectified linear unit (ReLU) activation function, 21, 209
recurrent neural networks (RNNs), 99–101, 103, 112. See also neural networks
reducing overfitting
  with data, 23–27, 209–210
  with model modifications, 29–36, 210
regression, conformal prediction and confidence intervals for, 178, 220
regularization, reducing overfitting
  with, 30–31, 36
reinforcement learning with human feedback (RLHF), 124
relative positional embeddings (relative positional encodings), 82, 85
ReLU (rectified linear unit) activation function, 21, 209
reparameterization, 151
representation learning, 11
representations, 3, 6–7
RepVGG CNN architecture, 151, 152
residual connection in transformer architecture, 107
ResNet-34 convolutional neural networks, 146, 217
resources, online, xxviii
retraining
  with different random seeds, 169–170
  stateless, 139–140, 141, 217
RLHF (reinforcement learning with human feedback), 124
RNNs (recurrent neural networks), 99–101, 103, 112. See also neural networks
RoBERTa (robustly optimized BERT approach), 108, 112
root mean square error (RMSE), 183, 220–221
root-squared error, 181
ROUGE (recall-oriented understudy for gisting evaluation) score, 128, 131–132, 133, 134
runtime algorithms as source of randomness, 61–62
S
SAINT method, 208
sample contrastive self-supervised learning, 14
sampling as source of randomness, 60, 65
sanity check, 189
scaled-dot product attention, 40, 42. See also self-attention mechanism
SCARF method, 208
score method of conformal prediction, 176–177, 178
SE (squared error) loss, 181
seeding random generator, 60, 61
self-attention mechanism.
  See also transformers
vs. Bahdanau attention
  mechanism, 99–101
overview, 99, 101–102
sequence parallelism, 40
transformers, 42, 43–45, 46, 47
in vision transformers, 83–84
self-prediction, 11–12
self-supervised learning
  contrastive, 12–14
encoder-only architectures, 108
leveraging unlabeled data, 11
limited labeled data, 194–195, 203,
  204, 221–222
overview, 9
pretraining transformers via, 45
reducing overfitting with, 25
self-prediction, 11–14
vs. transfer learning, 9–11
self-training, 199. See also knowledge
distillation
semi-supervised learning, 198–199, 203
sentence shuffling, 95
[SEP] token, 108
sequence parallelism, 40–41, 42
sequence-to-sequence (seq2seq)
  models, 107–110
sequential inference, 148
SGD (stochastic gradient descent)
  optimizer, 73, 212
shortcut connection, 107
siamese network setup, 13
similarity, embeddings as encoding, 5
  .632 bootstrap, 171
skip connection in transformer
  architecture, 107
skip-gram approach, Word2vec, 90
smaller models, reducing overfitting
  with, 31–33
soft attention, 211
soft parameter sharing, 200
soft prompting, 119–121, 125
sources of randomness
  dataset sampling and shuffling, 60
  different runtime algorithms, 61–62
  and generative AI, 62–64
  hardware and drivers, 62
  model weight initialization, 59–60
  nondeterministic algorithms, 61
  overview, 59
spatial attention, 215
spatial invariance, 80–82
speeding up inference. See model
  inference, speeding up
squared error (SE) loss, 181
Stable Diffusion latent diffusion
  model, 58
stacking (stacked generalization), 33
stateful training, 139, 140–141, 217
stateless training (stateless retraining),
  139–140, 141, 217
statistical population, 164
statistical two-sample tests, 218
stochastic diffusion process, 56
stochastic gradient descent (SGD)
  optimizer, 73, 212
stride, 78, 213
structured pruning, 20
student in knowledge distillation,
  31–32
supervised learning, 15. See also limited
  labeled data
support set in few-shot learning, 16
synonym replacement (text
  augmentation), 93–94
synthetic data generation, 96–97
T
TabNet, 208
tabular data, self-supervised learning
  for, 14, 208
teacher in knowledge distillation,
  31–32
10-fold cross-validation, 187, 188
TensorFlow framework, 59, 62, 149
tensor parallelism, 37, 38–40, 41–42, 211
test sets
  bootstrapping, 169, 170, 171
  conformal predictions, 176
discordance with training sets,
  189–191, 221
text, data augmentation for, 93–97,
  214–215
T5 encoder-decoder architecture, 112
time complexity of self-attention, 103, 215

top-k sampling, 63–64, 212

training. See also multi-GPU training paradigms; pretraining; randomness, sources of; retraining

epochs, tuning number of, 35, 210

post-training quantization, 151

procedure for lottery ticket hypothesis, 19–20

quantization-aware, 151

self-training, 199

stateless and stateful, 139–141, 217

training sets

conformal predictions, 176, 177
discordance with test sets, 189–191, 221
for vision transformers, 79–85, 213–214

transfer learning

limited labeled data, 194, 203, 204, 221–222
reducing overfitting with, 25, 26, 209

vs. self-supervised learning, 9–11

transformers. See also self-attention mechanism

adapting pretrained language models, 124–125

attention mechanism, 40, 43–45
classification tasks, 113–116
contemporary models, 111–112
decoders, 108–110
encoder-decoder hybrids, 110
encoders, 107–108
in-context learning, indexing, and prompt tuning, 116–119

multi-GPU training paradigms, 40, 42

number of parameters, 45
original architecture for, 105–110
overview, 105, 113
parallelization, 45–46
parameter-efficient fine-tuning, 119–124

pretraining via self-supervised learning, 45
reinforcement learning with human feedback (RLHF), 124
success of, 43–47
terminology, 110
transfer learning, 11

translation

back, 96
invariance and equivariance, 80–82
tokens, 44–46, 63–64, 102, 106–110, 117–118
triangle inequality, 180, 181, 182
ture generalization accuracy, 164
two-dimensional embeddings, 4–5
typo introduction, 95

U

unlabeled data in self-supervised learning, 10, 11
unstructured pruning, 20
unsupervised pretraining. See self-supervised learning

V

variational autoencoders (VAEs), 51–52, 53, 54, 57, 58
variational inference, 51
vectorization, 148–149, 152, 218
VideoBERT model, 201, 204
video data, applying self-supervised learning to, 14, 208
vision transformers (ViTs)

vs. convolutional neural networks, 79, 82–83, 84
inductive biases in, 83–84
large training sets for, 79
positional information in, 82, 85
recommendations for, 84

W

weakly supervised learning, 197–199, 203
weight decay, 30, 35
weighted loss function, 155
weight initialization, 59–60
weight normalization, 34–35
weight pruning, 19, 20
weights
  in convolutional layers, 70–71, 76–77
  in fully connected layers, 72, 76
weight sharing, 80, 81
winning tickets (lottery ticket hypothesis), 20, 21
Winograd-based convolution, 62, 65
word deletion (text augmentation), 94
word embeddings. See embeddings
WordNet thesaurus, 94, 97
word position swapping (word shuffling/permutation), 94–95
Word2vec model, 90, 92
X
XGBoost model, 26, 209
Z
zero-shot learning, 195–196. See also few-shot learning; in-context learning
z-scores (confidence intervals), 166