A. Related work

There have been several major lines of research on the theoretical understanding of neural networks. The first one deals with understanding the properties of the objective function used when training neural networks (Choromanska et al., 2014; Sagun et al., 2014; Zhang et al., 2015; Livni et al., 2014; Kawaguchi, 2016). The second involves studying the black-box optimization algorithms that are often used for training these networks (Hardt et al., 2015; Lian et al., 2015). The third analyzes the statistical and generalization properties of neural networks (Bartlett, 1998; Zhang et al., 2016; Neyshabur et al., 2015; Sun et al., 2016). The fourth adopts a generative point of view assuming that the data actually comes from a particular network, which it shows how to recover (Arora et al., 2014; 2015). The fifth investigates the expressive ability of neural networks, analyzing what types of mappings they can learn (Cohen et al., 2015; Eldan & Shamir, 2015; Telgarsky, 2016; Daniely et al., 2016). This paper is most closely related to the work on statistical and generalization properties of neural networks. However, instead of analyzing the problem of learning with a fixed architecture, we study a more general task of learning both architecture and model parameters simultaneously. On the other hand, the insights that we gain by studying this more general setting can also be directly applied to the setting with a fixed architecture.

There has also been extensive work involving structure learning for neural networks (Kwok & Yeung, 1997; Leung et al., 2003; Islam et al., 2003; Lehtokangas, 1999; Islam et al., 2009; Ma & Khorasani, 2003; Narasimha et al., 2008; Han & Qiao, 2013; Kotani et al., 1997; Alvarez & Salzmann, 2016). All these publications seek to grow and prune the neural network architecture using some heuristic. More recently, search-based approaches have been an area of active research (Ha et al., 2016; Chen et al., 2015; Zoph & Le, 2016; Baker et al., 2016). In this line of work, a learning meta-algorithm is used to search for an efficient architecture. Once a better architecture is found, previously trained networks are discarded. This search requires a significant amount of computational resources. Additionally, (Saxena & Verbeek, 2016) presented an approach to overcome the tedious process of exploring individual network architectures via a so-called fabric that embeds an exponentially large number of architectures.

To the best of our knowledge, none of these methods comes with a theoretical guarantee on their performance. Furthermore, optimization problems associated with these methods are intractable. In contrast, the structure learning algorithms introduced in this paper are directly based on data-dependent generalization bounds and aim to solve a convex optimization problem by adaptively growing the network and preserving previously trained components.

Finally, (Janzamin et al., 2015) is another paper that analyzes the generalization and training of two-layer neural networks through tensor methods. Our work uses different methods, applies to arbitrary networks, and also learns a network structure from a single input layer.

B. Proofs

We will use the following structural learning guarantee for ensembles of hypotheses.

Theorem 2 (DeepBoost Generalization Bound, Theorem 1, (Cortes et al., 2014))**.** *Let $\mathcal{H}$ be a hypothesis set admitting a decomposition $\mathcal{H} = \cup_{i=1}^{l} \mathcal{H}_i$ for some $l > 1$. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample S from D^m, the following inequality holds for any $f = \sum_{t=1}^{T} \alpha_t h_t$ with $\alpha_t \in \mathbb{R}_+$ and $\sum_{t=1}^{T} \alpha_t = 1$:*

$$R(f) \leq \widehat{R}_{S,\rho} + \frac{4}{\rho} \sum_{t=1}^{T} \alpha_t \mathfrak{R}_m(\mathcal{H}_{k_t}) + \frac{2}{\rho} \sqrt{\frac{\log l}{m}} + \sqrt{\left\lceil \frac{4}{\rho^2} \log\left(\frac{\rho^2 m}{\log l}\right) \right\rceil \frac{\log l}{m} + \frac{\log(\frac{2}{\delta})}{2m}},$$

where, for each $h_t \in \mathcal{H}$, k_t denotes the smallest $k \in [l]$ such that $h_t \in \mathcal{H}_{k_t}$.

Theorem 1. *Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample S of size m from $\mathcal{D}^m$, the following inequality holds for all $f = \sum_{k=1}^{l} \mathbf{w}_k \cdot \mathbf{h}_k \in \mathcal{F}$:*

$$R(f) \leq \widehat{R}_{S,\rho}(f) + \frac{4}{\rho} \sum_{k=1}^{l} \left\|\mathbf{w}_k\right\|_1 \mathfrak{R}_m(\widetilde{\mathcal{H}}_k) + \frac{2}{\rho} \sqrt{\frac{\log l}{m}} + C(\rho, l, m, \delta),$$

where $C(\rho, l, m, \delta) = \sqrt{\lceil \frac{4}{\rho^2} \log(\frac{\rho^2 m}{\log l}) \rceil \frac{\log l}{m} + \frac{\log(\frac{2}{\delta})}{2m}} = \widetilde{O}\left(\frac{1}{\rho}\sqrt{\frac{\log l}{m}}\right)$.

Proof. This result follows directly from Theorem 2. □

Theorem 1 can be straightforwardly generalized to the multi-class classification setting by using the ensemble margin bounds of Kuznetsov et al. (2014).

Lemma 1. *For any $k > 1$, the empirical Rademacher complexity of $\mathcal{H}_k$ for a sample S of size m can be upper-bounded as follows in terms of those of $\mathcal{H}_s$s with $s < k$:*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}_k) \leq 2 \sum_{s=1}^{k-1} \Lambda_{k,s} n_s^{\frac{1}{q}} \widehat{\mathfrak{R}}_S(\mathcal{H}_s).$$

Proof. By definition, $\widehat{\mathfrak{R}}_S(\mathcal{H}_k)$ can be expressed as follows:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}_k) = \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{\mathbf{h}_s \in \mathcal{H}_s^{n_s} \\ \|\mathbf{u}_s\|_p \leq \Lambda_{k,s}}} \sum_{i=1}^{m} \sigma_i \sum_{s=1}^{k-1} \mathbf{u}_s \cdot (\varphi_s \circ \mathbf{h}_s)(x_i)\right].$$

By the sub-additivity of the supremum, it can be upper-bounded as follows:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}_k) \leq \sum_{s=1}^{k-1} \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{\mathbf{h}_s \in \mathcal{H}_s^{n_s} \\ \|\mathbf{u}_s\|_p \leq \Lambda_{k,s}}} \sum_{i=1}^{m} \sigma_i \mathbf{u}_s \cdot (\varphi_s \circ \mathbf{h}_s)(x_i)\right].$$

We now bound each term of this sum, starting with the following chain of equalities:

$$\begin{aligned}
&\frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{\mathbf{h}_s \in \mathcal{H}_s^{n_s} \\ \|\mathbf{u}_s\|_p \leq \Lambda_{k,s}}} \sum_{i=1}^{m} \sigma_i \mathbf{u}_s \cdot (\varphi_s \circ \mathbf{h}_s)(x_i)\right] \\
&= \frac{\Lambda_{k,s}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{h}_s \in \mathcal{H}_s^{n_s}} \left\| \sum_{i=1}^{m} \sigma_i (\varphi_s \circ \mathbf{h}_s)(x_i)\right\|_q\right] \\
&= \frac{\Lambda_{k,s} n_s^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h \in \mathcal{H}_s} \left| \sum_{i=1}^{m} \sigma_i (\varphi_s \circ h)(x_i)\right|\right] \\
&= \frac{\Lambda_{k,s} n_s^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{h \in \mathcal{H}_s \\ \sigma \in \{-1,+1\}}} \sigma \sum_{i=1}^{m} \sigma_i (\varphi_s \circ h)(x_i)\right],
\end{aligned}$$

where the second equality holds by definition of the dual norm and the third equality by the following equality:

$$\begin{aligned}
\sup_{z_i \in Z} \|\mathbf{z}\|_q &= \sup_{z_i \in Z}\left[\sum_{i=1}^{n} |z_i|^q\right]^{\frac{1}{q}} = \left[\sum_{i=1}^{n} [\sup_{z_i \in Z} |z_i|]^q\right]^{\frac{1}{q}} \\
&= n^{\frac{1}{q}} \sup_{z_i \in Z} |z_i|.
\end{aligned}$$

The following chain of inequalities concludes the proof:

$$\begin{aligned}
&\frac{\Lambda_{k,s} n_s^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{h \in \mathcal{H}_s \\ \sigma \in \{-1,+1\}}} \sigma \sum_{i=1}^{m} \sigma_i (\varphi_s \circ h)(x_i)\right] \\
&\leq \frac{\Lambda_{k,s} n_s^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h \in \mathcal{H}_s} \sum_{i=1}^{m} \sigma_i (\varphi_s \circ h)(x_i)\right] \\
&\quad + \frac{\Lambda_{k,s}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h \in \mathcal{H}_s} \sum_{i=1}^{m} -\sigma_i (\varphi_j \circ h)(x_i)\right] \\
&= \frac{2\Lambda_{k,s} n_s^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h \in \mathcal{H}_s} \sum_{i=1}^{m} \sigma_i (\varphi_s \circ h)(x_i)\right] \\
&\leq \frac{2\Lambda_{k,s} n_s^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h \in \mathcal{H}_s} \sum_{i=1}^{m} \sigma_i h(x_i)\right] \\
&\leq 2\Lambda_{k,s} n_s^{\frac{1}{q}} \widehat{\mathfrak{R}}_S(\mathcal{H}_s),
\end{aligned}$$

where the second inequality holds by Talagrand's contraction lemma. □

Lemma 2. *Let $\Lambda_k = \prod_{s=1}^{k} 2\Lambda_{s,s-1}$ and $N_k = \prod_{s=1}^{k} n_{s-1}$. Then, for any $k \geq 1$, the empirical Rademacher complexity of $\mathcal{H}_k^*$ for a sample S of size m can be upper bounded as follows:*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}_k^*) \leq r_\infty \Lambda_k N_k^{\frac{1}{q}} \sqrt{\frac{\log(2n_0)}{2m}}.$$

Proof. The empirical Rademacher complexity of $\mathcal{H}_1$ can be bounded as follows:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\mathcal{H}_1) &= \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\|\mathbf{u}\|_p \leq \Lambda_{1,0}} \sum_{i=1}^{m} \sigma_i \mathbf{u} \cdot \boldsymbol{\Psi}(x_i)\right] \\
&= \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\|\mathbf{u}\|_p \leq \Lambda_{1,0}} \mathbf{u} \cdot \sum_{i=1}^{m} \sigma_i \boldsymbol{\Psi}(x_i)\right] \\
&= \frac{\Lambda_{1,0}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\| \sum_{i=1}^{m} \sigma_i [\boldsymbol{\Psi}(x_i)]\right\|_q\right] \\
&\leq \frac{\Lambda_{1,0} n_0^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\| \sum_{i=1}^{m} \sigma_i [\boldsymbol{\Psi}(x_i)]\right\|_\infty\right] \\
&= \frac{\Lambda_{1,0} n_0^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\max_{j \in [1,n_1]} \left| \sum_{i=1}^{m} \sigma_i [\boldsymbol{\Psi}(x_i)]_j\right|\right] \\
&= \frac{\Lambda_{1,0} n_0^{\frac{1}{q}}}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\max_{\substack{j \in [1,n_1] \\ s \in \{-1,+1\}}} \sum_{i=1}^{m} \sigma_i s[\boldsymbol{\Psi}(x_i)]_j\right] \\
&\leq \Lambda_{1,0} n_0^{\frac{1}{q}} r_\infty \sqrt{m} \frac{\sqrt{2\log(2n_0)}}{m} \\
&= r_\infty \Lambda_{1,0} n_0^{\frac{1}{q}} \sqrt{\frac{2\log(2n_0)}{m}}.
\end{aligned}$$

The result then follows by application of Lemma 1. □

Corollary 1. *Fix $\rho > 0$. Let $\Lambda_k = \prod_{s=1}^{k} 4\Lambda_{s,s-1}$ and $N_k = \prod_{s=1}^{k} n_{s-1}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of a sample S of size m from $\mathcal{D}^m$, the following inequality holds for all $f = \sum_{k=1}^{l} \mathbf{w}_k \cdot \mathbf{h}_k \in \mathcal{F}^*$:*

$$\begin{aligned}
R(f) \leq \widehat{R}_{S,\rho}(f) &+ \frac{2}{\rho}\sum_{k=1}^{l} \|\mathbf{w}_k\|_1 \left[\overline{r}_\infty \Lambda_k N_k^{\frac{1}{q}} \sqrt{\frac{2\log(2n_0)}{m}}\right] \\
&+ \frac{2}{\rho}\sqrt{\frac{\log l}{m}} + C(\rho, l, m, \delta),
\end{aligned}$$

where $C(\rho, l, m, \delta) = \sqrt{\left\lceil \frac{4}{\rho^2} \log\left(\frac{\rho^2 m}{\log l}\right)\right\rceil \frac{\log l}{m} + \frac{\log\left(\frac{2}{\delta}\right)}{2m}} = \widetilde{O}\left(\frac{1}{\rho}\sqrt{\frac{\log l}{m}}\right)$, and where $\overline{r}_\infty = \mathbb{E}_{S \sim \mathcal{D}^m}[r_\infty]$.

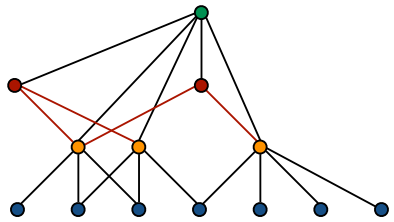

Figure 4. Illustration of a neural network designed by AdaNet.CVX. Units at each layer (other than the output layer) are only connected to units in the layer below.

Proof. Since $\mathcal{F}^*$ is the convex hull of $\mathcal{H}^*$, we can apply Theorem 1 with $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k^*)$ instead of $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$. Observe that, since for any $k \in [l]$, $\widetilde{\mathcal{H}}_k^*$ is the union of $\mathcal{H}_k^*$ and its reflection, to derive a bound on $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k^*)$ from a bound on $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$ it suffices to double each $\Lambda_{s,s-1}$. Combining this observation with the bound of Lemma 2 completes the proof. □

C. Alternative algorithm

In this section, we present an alternative algorithm, AdaNet.CVX, that generates candidate subnetworks in closed-form using Banach space duality.

As in Section 5, let f_{t-1} denote the AdaNet model after $t-1$ rounds, and let l_{t-1} be the depth of the architecture. AdaNet.CVX will consider $l_{t-1}+1$ candidate subnetworks, one for each layer in the model plus an additional one for extending the model.

Let $h^{(s)}$ denote the candidate subnetwork associated to layer $s \in [l_{t-1}+1]$. We define $h^{(s)}$ to be a single unit in layer s that is connected to units of f_{t-1} in layer $s-1$:

$$h^{(s)} \in \{x \mapsto \mathbf{u} \cdot (\varphi_{s-1} \circ \mathbf{h}_{s-1,t-1})(x) : \mathbf{u} \in \mathbb{R}^{n_{s-1,t-1}}, \|\mathbf{u}\|_p \leq \Lambda_{s,s-1}\}.$$

See Figure 4 for an illustration of the type of neural network designed using these candidate subnetworks.

For convenience, we denote this space of subnetworks by $\mathcal{H}'_s$:

$$\mathcal{H}'_s = \{x \mapsto \mathbf{u} \cdot (\varphi_{s-1} \circ \mathbf{h}_{s-1,t-1})(x) : \mathbf{u} \in \mathbb{R}^{n_{s-1,t-1}}, \|\mathbf{u}\|_p \leq \Lambda_{s,s-1}\}.$$

Now, recall the notation

$$F_t(w,h) = \frac{1}{m}\sum_{i=1}^m \Phi\Big(1 - y_i(f_{t-1}(x_i) - wh(x_i))\Big) + \Gamma_h|w|$$

used in Section 5. As in AdaNet, the candidate subnetwork chosen by AdaNet.CVX is given by the following optimization problem:

$$\operatorname*{argmin}_{h \in \cup_{s=1}^{l_{t-1}+1} \mathcal{H}'_s} \min_{w \in \mathbb{R}} F_t(w,h).$$

Remarkably, the subnetwork that solves this infinite dimensional optimization problem can be obtained directly in closed-form:

Theorem 3 (AdaNet.CVX Optimization)**.** *Let (w^*, h^*) be the solution to the following optimization problem:*

$$\operatorname*{argmin}_{h \in \cup_{s=1}^{l_{t-1}} \mathcal{H}'_s} \min_{w \in \mathbb{R}} F_t(w,h).$$

Let D_t be a distribution over the sample $(x_i, y_i)_{i=1}^m$ such that $D_t(i) \propto \Phi'(1 - y_i f_{t-1}(x_i))$, and denote $\epsilon_{t,h} = \mathbb{E}_{i \sim D_t}[y_i h(x_i)]$.

Then,

$$w^* h^* = w^{(s^*)} h^{(s^*)},$$

where $(w^{(s^*)}, h^{(s^*)})$ are defined by:

$$s^* = \operatorname*{argmax}_{s \in [l_t - 1]} \Lambda_{s,s-1} \|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q.$$

$$u_i^{(s)} = \frac{\Lambda_{s,s-1} |\epsilon_{t,h_{s-1,t-1,i}}|^{q-1} \operatorname{sgn}(\epsilon_{t,h_{s-1,t-1,i}})}{\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q^{\frac{q}{p}}}$$

$$h^{(s^*)} = \mathbf{u}^{(s^*)} \cdot (\varphi_s \circ \mathbf{h}_{s-1,t-1})$$

$$w^{(s^*)} = \operatorname*{argmin}_{w \in \mathbb{R}} \frac{1}{m}\sum_{i=1}^m \Phi\Big(1 - y_i f_{t-1}(x_i) - y_i w h^{(s^*)}(x_i)\Big) + \Gamma_{s^*}|w|.$$

Proof. By definition,

$$F_t(w,h) = \frac{1}{m}\sum_{i=1}^m \Phi\Big(1 - y_i\big(f_{t-1}(x_i) - wh(x_i)\big)\Big) + \Gamma_h|w|.$$

Notice that the minimizer over $\cup_{s=1}^{l_{t-1}+1} \mathcal{H}'_s$ can be determined by comparing the minimizers over each $\mathcal{H}'_s$.

Moreover, since the penalty term $\Gamma_h|w|$ has the same contribution for every $h \in \mathcal{H}'_s$, it has no impact on the optimal choice of h over $\mathcal{H}'_s$. Thus, to find the minimizer over each $\mathcal{H}'_s$, we can compute the derivative of $F_t - \Gamma_h|w|$ with respect to w:

$$\frac{d(F_t - \Gamma_h|\eta|)}{dw}(w,h) = \frac{-1}{m}\sum_{i=1}^m y_i h(x_i) \Phi'\Big(1 - y_i f_{t-1}(x_i)\Big).$$

Now, if we let

$$D_t(i)S_t = \Phi'\Big(1 - y_i f_{t-1}(x_i)\Big),$$

then this expression is equal to

$$-\left[\sum_{i=1}^{m} y_i h(x_i) D_t(i)\right]\frac{S_t}{m} = (2\epsilon_{t,h} - 1)\frac{S_t}{m},$$

where $\epsilon_{t,h} = \mathbb{E}_{i\sim D_t}[y_i h(x_i)]$. Thus, it follows that for any $s \in [l_{t-1}+1]$,

$$\operatorname*{argmax}_{h\in\mathcal{H}'_s} \frac{d(F_t - \Gamma_h|w|)}{dw}(w,h) = \operatorname*{argmax}_{h\in\mathcal{H}'_s} \epsilon_{t,h}.$$

Note that we still need to search for the optimal descent coordinate over an infinite dimensional space. However, we can write

$$\begin{aligned}
&\max_{h\in\mathcal{H}'_s} \boldsymbol{\epsilon}_{t,h} \\
&= \max_{h\in\mathcal{H}'_s} \mathop{\mathbb{E}}_{i\sim D_t}[y_i h(x_i)] \\
&= \max_{\mathbf{u}\in\mathbb{R}^{n_{s-1,t-1}}} \mathop{\mathbb{E}}_{i\sim D_t}[y_i \mathbf{u}\cdot(\varphi_{s-1}\circ\mathbf{h}_{s-1,t-1})(x_i)] \\
&= \max_{\mathbf{u}\in\mathbb{R}^{n_{s-1,t-1}}} \mathbf{u}\cdot \mathop{\mathbb{E}}_{i\sim D_t}[y_i(\varphi_{s-1}\circ\mathbf{h}_{s-1,t-1})(x_i)].
\end{aligned}$$

Now, if we denote by $u^{(s)}$ the connection weights associated to $h^{(s)}$, then we claim that

$$u_i^{(s)} = \frac{\Lambda_{s,s-1}|\epsilon_{t,h_{s-1,t-1,i}}|^{q-1}\operatorname{sgn}(\epsilon_{t,h_{s-1,t-1,i}})}{\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q^{\frac{q}{p}}},$$

which is a consequence of Banach space duality. To see this, note first that by Hölder's inequality, every $\mathbf{u} \in \mathbb{R}^{n_{s-1,t-1}}$ with $\|\mathbf{u}\|_p \le \Lambda_{s,s-1}$ satisfies:

$$\begin{aligned}
&\mathbf{u}\cdot \mathop{\mathbb{E}}_{i\sim D_t}[y_i(\varphi_{s-1}\circ\mathbf{h}_{s-1,t-1})(x_i)] \\
&\le \|\mathbf{u}\|_p \Big\| \mathop{\mathbb{E}}_{i\sim D_t}[y_i(\varphi_{s-1}\circ\mathbf{h}_{s-1,t-1})(x_i)]\Big\|_q \\
&\le \Lambda_{s,s-1}\Big\| \mathop{\mathbb{E}}_{i\sim D_t}[y_i(\varphi_{s-1}\circ\mathbf{h}_{s-1,t-1})(x_i)]\Big\|_q.
\end{aligned}$$

At the same time, our choice of $\mathbf{u}^{(s)}$ also attains this upper bound:

$$\begin{aligned}
&\mathbf{u}^{(s)}\cdot\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}} \\
&= \sum_{i=1}^{n_{s-1,t-1}} u_i^{(s)}\epsilon_{t,h_{s-1,t-1,i}} \\
&= \sum_{i=1}^{n_{s-1,t-1}} \frac{\Lambda_{s,s-1}}{\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q^{\frac{q}{p}}}|\epsilon_{t,h_{s-1,t-1,i}}|^q \\
&= \frac{\Lambda_{s,s-1}}{\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q^{\frac{q}{p}}}\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q^q \\
&= \Lambda_{s,s-1}\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q.
\end{aligned}$$

ADANET.CVX$(S = ((x_i, y_i)_{i=1}^m)$

$f_0 \leftarrow 0$
for $t \leftarrow 1$ **to** T **do**
  $s^* \leftarrow \operatorname{argmax}_{s\in[l_{t-1}+1]} \Lambda_{s,s-1}\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q$.
  $u_i^{(s^*)} \leftarrow \frac{\Lambda_{s^*,s^*-1}|\epsilon_{t,h_{s^*-1,t-1,i}}|^{q-1}\operatorname{sgn}(\epsilon_{t,h_{s-1,t-1,i}})}{\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s^*-1,t-1}}\|_q^{\frac{q}{p}}}$
  $\mathbf{h}' \leftarrow \mathbf{u}^{(s^*)}\cdot(\phi_{s^*-1}\circ\mathbf{h}_{s^*-1,t-1})$
  $\eta' \leftarrow$ MINIMIZE$(\tilde{F}_t(\eta, \mathbf{h}'))$
  $f_t \leftarrow f_{t-1} + \eta'\cdot\mathbf{h}'$
return f_T

Figure 5. Pseudocode of the ADANET.CVX algorithm.

Thus, $\mathbf{u}^{(s)}$ and the associated network $h^{(s)}$ is the coordinate that maximizes the derivative of F_t with respect to w among all subnetworks in $\mathcal{H}'_s$. Moreover, $h^{(s)}$ also achieves the value: $\Lambda_{s,s-1}\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q$.

This implies that by computing $\Lambda_{s,s-1}\|\boldsymbol{\epsilon}_{t,\mathbf{h}_{s-1,t-1}}\|_q$ for every $s \in [l_{t-1}+1]$, we can find the descent coordinate across all $s \in [l_{t-1}+1]$ that improves the objective by the largest amount. Moreover, we can then solve for the optimal step size in this direction to compute the weight update. □

The theorem above defines the choice of descent coordinate at each round and motivates the following algorithm, ADANET.CVX. At each round, ADANET.CVX can design the optimal candidate subnetwork within its searched space in closed form, leading to an extremely efficient update. However, this comes at the cost of a more restrictive search space than the one used in ADANET. The pseudocode of ADANET.CVX is provided in Figure 5.

D. More experiments

In this section, we report the results of some additional experiments.

In our first set of experiments, we compared how many trials were needed for a bandit algorithm (Snoek et al., 2012) to find a close-to-optimal set of hyperparameters. Our experiment (see Table 5) shows that both ADANET and traditional NNs often find close-to-optimal parameter values within the first 200 trials. The number of trials were averaged over 10 folds. Note that optimal parameters for ADANET result in better accuracy than those of traditional NNs. This serves as further evidence that ADANET is more efficient in both finding close-to-optimal accuracy values and in finding the best optimal network architecture when compared to traditional NNs.

It should be noted that, in general, the number of trials that

Table 5. Performance of hyperparameter search on the CIFAR `cat-dog` task.

Algorithm	Average number of trials
ADANET	165.8
NN	136

are needed to find close-to-optimal parameters depends on the algorithm used to perform hyperparameter optimization.

Our final experiment consisted of first running the ADANET algorithm to learn an architecture and weights and then running back-propagation algorithm on the resulting architecture with weights learned by ADANET as initialization. We used the `cat-dog` label pair task and $3{,}000$ back-propagation steps with the same learning rate as the one used to train each subnetwork. Using the same cross-validation setup as in Section 6 for our CIFAR-10 experiments, this led to a test accuracy of 0.6908 (with a standard deviation of 0.01224), which is slightly worse than the accuracy of 0.6924 obtained by running the ADANET algorithm alone. This further demonstrates that ADANET is able to learn both the network architecture and its weights simultaneously.

E. Implementation details

In this section, we provide some additional implementation details regarding the experiments presented in Section 6. Our system involves two components implemented on a CPU: 1) a subnetwork model that handles each subnetwork; 2) an ADANET model to compose multiple subnetworks and a classification layer to combine all output weights. For both these components, we used the Tensorflow package. We implemented a custom layer using matrix multiplication, including embeddings from the other subnetworks. In addition, our loss function was based on Eq. (5) and optimized via stochastic optimization (Kingma & Ba, 2014).